Modelling and Compensation Techniques for Short Duration Speaker Verification

Jianbo Ma

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Electrical Engineering and Telecommunications

Faculty of Engineering

UNSW Sydney

January 2019

UNSW SYDNEY | Australia's Global University

Thesis/Dissertation Sheet

Surname/Family Name	:	**Ma**
Given Name/s	:	**Jianbo**
Abbreviation for degree as give in the University calendar	:	**PHD**
Faculty	:	**Faculty of Engineering**
School	:	**School of Electrical Engineering and Telecommunications**
Thesis Title	:	**Modelling and compensation techniques for short duration speaker verification**

Abstract 350 words maximum: (PLEASE TYPE)

Voice based biometric systems have been the focus of active research for a number of decades. These systems have a number of advantages including their non-invasive nature and the ability to transmit voice over a variety of channels. However, a key difficulty that impedes their wide-spread use is the inability of current systems to work accurately with short speech utterances since it is unrealistic to collect long utterances in many scenarios. The goal of this thesis is to improve the accuracy of speaker verification system on utterances that are less than ten seconds long.

This thesis shows that the conventional i-vector representation, derived from the total variability model, is not an accurate representation for short duration utterances. More accurate models are developed in this thesis. Firstly, a generalization of the total variability model is developed by allowing the distribution of latent variables to be mixture of Gaussians. Secondly, it was found that the information in each phonetic group, referred to as the local acoustic variability, is complementary to the total variability model. Consequently, this thesis proposes a local acoustic model that utilises this information. Thirdly, the current i-vector representation of an utterance is sensitive to phonetic mismatch, which is severe in short utterances. This thesis proposes a mixture of total variability models to have speaker-phonetic vector representations.

The vectors representing short utterances are also distributed differently to those representing long utterances, and this difference will propagate into the back-end. As such, this thesis proposes compensation techniques. Specifically, projection methods based on a Gaussian probabilistic linear discriminant analysis (GPLDA) model with tied latent variables, and neural networks are proposed to normalise duration mismatches in the vector representation space. In the projected space, vector representations of long and short utterances are more likely to be similarly distributed. Finally, a twin model GPLDA back-end that uses two different sets of parameters to model short and long utterances differently, connected by shared speaker identity, is proposed to generate more reliable scores.

Modelling and compensation techniques proposed in this thesis are efficient to mitigate problems caused by short duration in speaker verification.

FOR OFFICE USE ONLY Date of completion of requirements for Award:

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed .....

Date.....22/01/2019.....

COPYRIGHT STATEMENT

'I hereby grant the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstract International (this is applicable to doctoral theses only).

I have either used no substantial portions of copyright material in my thesis or I have obtained permission to use copyright material; where permission has not been granted I have applied/will apply for a partial restriction of the digital copy of my thesis or dissertation.'

Signed .....

Date ..... 22/01/2019 .....

AUTHENTICITY STATEMENT

'I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.'

Signed .....

Date ..... 22/01/2019 .....

ACKNOWLEDGEMENTS

This thesis concludes three and half years of my PhD study in UNSW Sydney. During this journey, I was so lucky to meet a lot of amazing people who have helped me, inspired me, and supported me.

First of all, I want to express my deepest gratitude to my supervisors, Prof. Eliathamby Ambikairajah and Dr. Vidhyasharan Sethu, for their immense knowledge, encouragement, patient academic guidance and continuous support. Their supervisions always encourage me to find ways when there is a difficulty and inspire more insightful understanding of problems. Their academic attitude influenced me to pursue high standard work in my PhD study. I could not have imagined having better supervisors for my PhD study.

Next, I thank Dr. Kong Aik Lee (NEC, Japan), who inspired me a lot in my PhD study and always support my work. I would like to thank that I4U team organised by Dr. Kong Aik Lee in NIST SRE 2016, from which I learnt a lot. I am also grateful to A/prof. Julien Epps, for whom gave a lot of advices during group meetings. I also would like to thank the members of my review panel, Prof. David Taubman, Dr. Ediz Cetin, Dr. Elias Aboutanios for their insightful comments and great advices.

I would like to thank Dr. Stefanie Brown for proofreading, and my colleagues at the UNSW Speech Signal Processing research group: Ting Dang, Kalani Wataraka Gamage and Kaavya Sriskandaraja, Saad Irtza, Sarith Fernando, Zhaocheng Huang, Brian Stasak, Dr. Siyuan Chen, Dr Phu Ngoc Le, for discussion; Hang Li, Gajan Suthokumar, Tharshini Gunendradasan for fun and emotional support. I would also thank Dr. Zhan Shi for her encouragement; Dr. Zeyu Li for his emotional support; Matthew King, Noah, Dr. Wei Wang for the joy and friendship.

I acknowledge the following sources of funding which have enabled me to pursue my PhD; UNSW Sydney for a Tuition Fee Scholarship, and CSIRO, Data61 for a Research Postgraduate Award. I thank the School of Electrical Engineering and Telecommunications UNSW for supporting me throughout my studies.

I would like to express my special appreciation to my soulmate Liujia Liu. Thanks for your strong belief and supports when I felt down.

Finally, I would like to express my heartfelt gratitude to my parents, my sister, and my grandparents. I am so blessed to have such a wonderful family that always believe me, trust me, and support me whenever I need them.

ABSTRACT

Voice based biometric systems have been the focus of active research for a number of decades. These systems have a number of advantages including their non-invasive nature and the ability to transmit voice over a variety of channels. However, a key difficulty that impedes their wide-spread use is the inability of current systems to work accurately with short speech utterances since it is unrealistic to collect long utterances in many scenarios. The goal of this thesis is to improve the accuracy of speaker verification system on utterances that are less than ten seconds long.

This thesis shows that the conventional i-vector representation, derived from the total variability model, is not an accurate representation for short duration utterances. More accurate models are developed in this thesis. Firstly, a generalization of the total variability model is developed by allowing the distribution of latent variables to be mixture of Gaussians. Secondly, it was found that the information in each phonetic group, referred to as the local acoustic variability, is complementary to the total variability model. Consequently, this thesis proposes a local acoustic model that utilises this information. Thirdly, the current i-vector representation of an utterance is sensitive to phonetic mismatch, which is severe in short utterances. This thesis proposes a mixture of total variability models to have speaker-phonetic vector representations.

The vectors representing short utterances are also distributed differently to those representing long utterances, and this difference will propagate into the back-end. As such, this thesis proposes compensation techniques. Specifically, projection methods based on a Gaussian probabilistic linear discriminant analysis (GPLDA) model with tied latent variables, and neural networks are proposed to normalise duration mismatches in the vector representation space. In the projected space, vector representations of long and short utterances are more likely to be

similarly distributed. Finally, a twin model GPLDA back-end that uses two different sets of parameters to model short and long utterances differently, connected by shared speaker identity, is proposed to generate more reliable scores.

Modelling and compensation techniques proposed in this thesis are efficient to mitigate problems caused by short duration in speaker verification.

ACRONYMS AND ABBREVIATIONS

ASV	Automatic Speaker Verification
ASR	Automatic Speech Recognition
CSD	Cosine Similarity Distance
CMS	Cepstral Mean Subtraction
CMVN	Cepstral Mean and Variance Normalization
DCT	Discrete Cosine Transform
DCF	Decision Cost Function
DNN	Deep Neural Networks
EER	Equal Error Rate
EM	Expectation Maximization
FAR	False Acceptance Rate
FM	Feature Mapping
FFT	Fast Fourier Transform
FW	Feature Warping
FA	Factor Analysis
GMM	Gaussian Mixture Model
GPLDA	Gaussian PLDA
HMM	Hidden Markov Models
HTPLDA	Heavy-tailed PLDA
IFA	Independent Factor Analysis
JFA	Joint Factor Analysis
KL	Kullback-Leibler
LDA	Linear Discriminant Analysis
LFCC	Linear Frequency Cepstral Coefficients
LPCC	Linear Prediction Cepstral Coefficients
LVM	Local Variability Model
LSTM	Long Short-Term Memory
MFCC	Mel Frequency Cepstral Coefficients
MAP	Maximum-A-Posteriori
ML	Maximum Likelihood

MinDCF	Minimum Detection Cost Function
MR	Missing Rate
MoG	Mixture of Gaussians
MTVM	Mixture of TVM
NAP	Nuisance Attribute Projection
NIST	National Institute of Science and Technology
PLDA	Probabilistic Linear Discriminant Analysis
PLPCC	Perceptual Linear Prediction Cepstral Coefficients
PPCA	Probabilistic Principal Component Analysis
QMF	Quality Measure Function
ReLU	Rectified Linear Unit
RS	Relative Spectral
SVM	Support Vector Machine
SRE	Speaker Recognition Evaluation
T-norm	Test Normalization
TD	Text-Dependent
TI	Text-Independent
TVM	Total Variability Model
TM-GPLDA	Twin model GPLDA
UBM	Universal Background Model
VQ	Vector Quantization
VAD	Voice Activity Detection
WCCN	Within-class Covariance Normalization
z-norm	Zero Normalization
zt-norm	Zero Test Normalization

TABLE OF CONTENTS

Acknowledgements I

Abstract III

Acronyms and Abbreviations V

Table of Contents VII

List of Figures XII

List of Tables XVI

1 Introduction 1

1.1 Research Overview 1

1.2 Short duration speaker verifications 3

1.3 Research targets 5

1.4 Thesis structure 6

1.5 List of contributions 8

1.6 List of publications 9

2 Literature Review 11

2.1 Automatic speaker verification system-overview 12

2.2 Front-end of automatic speaker verification system 14

2.2.1 Feature extraction 16

2.2.2 Feature normalisation 17

2.3 Back-end of automatic speaker verification systems 18

2.3.1 GMM-UBM system 18

2.3.2 Supervector with Support vector machines 22
2.3.3 Total variability model with probabilistic linear analysis model 23
2.3.4 Neural network based models 29
2.4 Short duration speaker verification 31
2.5 Database 36
2.6 Performance Evaluation 37
2.7 Summary 38
3 Parallel Speaker and Content Modelling for Text-dependent Speaker Verification 39
3.1 Modelling the alternative hypothesis 39
3.2 Proposed parallel system 40
3.2.1 Proposed speaker verification sub-system 40
3.2.2 Proposed lexical content sub-system 41
3.2.3 Score interpretation and combination 43
3.3 Mixture selection 44
3.4 Experiments and results 46
3.4.1 Parallel speaker and content modelling systems 46
3.4.2 Incorporating mixture selection 48
3.5 Summary 49
4 Model Compensation for Short Duration Speaker Verification 51
4.1 Analysis of short duration utterance in the i-vector space 51
4.2 Proposed generalised variability model 54
4.2.1 Mixture of Gaussians distribution as prior 55

4.2.2 Posterior inferences 57
4.2.3 Parameter estimation 60
4.3 Incorporating local acoustic information into the total variability model 61
4.3.1 Proposed local acoustic variability model 63
4.3.2 Likelihood weighting 65
4.3.3 Mean vector weighting 66
4.3.4 Score weighting 67
4.4 Experiments 67
4.4.1 Experiments and discussion of GVM 67
4.4.2 Experiments of local acoustic model 71
4.5 Summary 75
5 Speaker-Phonetic Vector Representation for Short Duration Utterance 77
5.1 Phonetic variability in the i-vector space for short utterances 78
5.2 Revising GVM to generate phonetic-speaker vectors 80
5.3 Proposed mixtures of the total variability model 81
5.3.1 Calculate variational posterior probability 85
5.3.2 Calculating a lower bound for VBEM 87
5.3.3 Parameter update formula 88
5.3.4 Tying parameters in mixture of total variability model 88
5.3.5 Scoring method 90
5.3.6 Discussion of mixtures of total variability model 90
5.4 Experimental evaluation of phonetic-speaker vector represen-tation 91

5.5 Summary 95

6 Duration Mismatch in Utterance Vector Representation Space 97

6.1 Analysis of duration mismatch in the vector representation space 98

6.2 Duration compensation with linear projection 101

6.2.1 Proposed Twin-Model projection method 102

6.3 Duration compensation with neural networks 105

6.3.1 Duplet centre loss 106

6.4 Experiments 109

6.4.1 Experiments with Twin-Model projection 109

6.4.2 Experiments with non-linear projection 111

6.5 Summary 116

7 Duration Mismatch Compensation in the Back-end 117

7.1 Duration mismatch analysis in pre-processed vector space 118

7.2 Compensating duration mismatch in back-ends 120

7.2.1 Proposed Twin Model GPLDA 120

7.2.2 Twin Model GPLDA Parameter Estimation 124

7.2.3 Incorporating uncertainty propagation 126

7.3 Experiments 133

7.3.1 Experiments with Twin Model GPLDA without uncertainty propagation 133

7.3.2 Experiments for uncertainty propagation 137

7.4 Summary 140

8 Conclusions and Future Work 143

8.1 Conclusions 143

8.1.1 Proposal of Parallel Speaker and Content Modelling for Text-dependent Speaker Verification 144

8.1.2 Analysis of total variability model for short duration utterances 145

8.1.3 Generalised variability model and the complementary local acoustic variability model 146

8.1.4 Speaker-phonetic vector representation of utterance 147

8.1.5 Mismatch compensation techniques for short duration 148

8.2 Future perspectives 150

Appendix A: Expectation Maximization Algorithm 153

Appendix B: EM Algorithm for GMM Training 157

9 References 160

LIST OF FIGURES

Figure 1.1 Speaker verification system performance over duration. 5

Figure 2.1 Diagram of a basic automatic speaker verification system, showing phases of model creation and speaker enrolment(top) and verification (bottom). 13

Figure 2.2 The different levels of speaker discriminative features [8]. 15

Figure 2.3 Extraction process of Mel-frequency cepstral coefficients (MFCCs). 17

Figure 2.4 A diagram of a conventional GMM-UBM system [49]. 21

Figure 2.5 A diagram of Support Vector Machine process, showing the hyperplane separating two classes, and the closest support vectors that are used to define the hyperplane [56]. 23

Figure 2.6 Diagram of the creation of a supervector from a set of GMM speaker models, and its conversion to the i-vector representation. 25

Figure 2.7 Graphical model representation of a total variability model. The variables are: z - labelling variables; x - feature frames; μ - means of the supervectors; ω - latent variable. The indices are: superscript i – utterance index; subscripts c - mixture component in UBM, and n - feature frame index. 25

Figure 2.8 The hypotheses that (a) the test and enrolment i-vectors are from same speaker (i.e share latent variables h), and (b) that enrolment and test i-vectors are from different speakers (i.e. have distinct latent variables h_e and h_t). 28

Figure 2.9 Graphical model representation of i-vector/GPLDA system, where $\theta_g g$ and θ_u denote the parameters of GPLDA and UBM respectively, E is the total number of enrolment data, the super script i denotes parameter or variables are from enrolment data, t denotes parameter or variables are from test data. (a) Hypothesis that test and enrolment i-vectors are from same speaker (share latent variables - h); (b) Hypothesis that test and enrolment i-vectors are from different speakers (distinct latent variables - $h_e \; and \; h_t$) 29

Figure 2.10 Detection error trade-off and Equal Error Rate. 38

Figure 2.11 Confusion matrix of ASV. 38
Figure 3.1 Proposed parallel speaker and content modelling. 40
Figure 3.2 Proposed speaker verification sub-system using HMMs. 41
Figure 3.3 Proposed lexical content sub-system using segment models. 42
Figure 4.1 Comparision between TVM and GVM. 56
Figure 4.2 Graphical model of the proposed generalized variability model. The variables are: q - state variables. The indexes are: k - state index. The remaining symbols are the same as in Figure 2.12. 57
Figure 4.3 Comparison between i-vector/GPLDA system (upper panel) and GPLDA in supervector space system (lower panel). 65
Figure 4.4 DET plot of two conditions a) CORE-EXT, and b) CORE-10SEC 71
Figure 4.5 DET plot of systems (i-vector/GPLDA, SLM_W, and fusion) for (a) 8CONV-10SEC, Male, (b) 8CONV-5SEC, Male, (c) 8CONV-3SEC, Male, (d) 8CONV-10SEC, Female, (e) 8CONV-5SEC, Female, and (f) 8CONV-3SEC, Female. 74
Figure 4.6 DET plot of systems (GVM, SLM_W, and fusion) for (a) 8CONV-10SEC, Male, (b) 8CONV-5SEC, Male, (c) 8CONV-3SEC, Male, (d) 8CONV-10SEC, Female, (e) 8CONV-5SEC, Female, (f) 8CONV-3SEC, Female. 75
Figure 5.1 Demonstration of phonetic i-vector clustering in two-dimension space. 79
Figure 5.2 Difference between (a) GVM, and (b) revised GVM to obtain phonetic-speaker vectors to represent on utterance. 81
Figure 5.3 Illustration of mixture of total variability model. 82
Figure 5.4 Graphical model representation of a total variability model. The variables are: z - labeling variables; x - feature frames; μ - means of the supervectors; ω - latent variable. The indexes are: superscript i – utterance index; subscripts c - mixture component in UBM, and n - feature frame index. 83

Figure 5.5 Graphical model of mixture of total variability model. The variables are: z - labelling variables; x - feature frames; μ - means of the supervectors; ω - latent variable; K – phonetic class labeling variables. The indexes are: superscript i – utterance index; subscripts c - mixture component in UBM, k – phonetic class index, and n - feature frame index. 83
Figure 6.1 Three-dimensional LDA visualization of i-vectors from long and short utterances of different speakers for (a) i-vectors from long utterances, and (b) i-vectors from short utterances. 100
Figure 6.2 Three-dimensional LDA visualization of i-vectors from long and short utterances of different speakers for (a) i-vectors from first speaker, and (b) i-vectors from second speaker.101
Figure 6.3 Graphical model of (a) standard GPLDA, and (b) Twin-Model projection. 102
Figure 6.4 Illustration of the required compensation effect, showing (a) a 2-dimensional representation of vector representations of utterances from multiple speakers, both long and short utterances, before compensation, and (b) after compensation. 107
Figure 6.5 Three-dimensional LDA visualization of i-vectors from long and short utterances of the same speakers for (a) i-vectors before transformation (b) i-vectors after transformation.. 110
Figure 6.6 Structure of a neural network for duration mismatch compensation. 112
Figure 6.7 Training and validation loss. 114
Figure 6.8 Two-dimensional visualisation of vector representations from different speakers (a) before training and (b) after training. 114
Figure 7.1 Histograms of i-vector lengths (magnitudes) estimated from long and short duration utterances (denoted as long and short i-vectors). 119
Figure 7.2 Graphical model of (a) standard GPLDA, and (b) Twin Model GPLDA. 121
Figure 7.3 Graphical models of the two hypotheses: (a) H_s that the test and enrolment i-vectors are from same speaker (i.e. share latent variables h); and (b) H_d that the test and enrolment i-vectors are from different speakers (i.e. have distinct latent variables h_1 and h_2). 122

Figure 7.4 Graphical models showing the hypotheses (a) the test and enrolment i-vectors are from same speake, i.e., share latent variables *y*; and (b) that the test and enrolment i-vectors are from different speakers, i.e., have distinct latent variables, h_e and h_t. 132
Figure 7.5 Performance (MinDCF %) using standard and the Twin-Model GPLDA systems on the NIST SRE'10 8CONV-10SEC, and additional 5SEC and 3SEC conditions (male part)... 135
Figure 7.6 Performance (MinDCF %) using standard and the Twin-Model GPLDA systems on the NIST SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions (female part) . 137
Figure A.0.1 An illustration of the likelihood decomposition equation (2-4) [50]. 154
Figure B.0.1 Graphical representation of a Gaussian mixture model [50], where x_n is the observed data, z_n is the latent variables, π is the mixing coefficient, μ is the mean, Σ_{UBM} is the covariance, and N is the number of data points. 159

LIST OF TABLES

Table 1.1 Performance (EER %) of the i-vector/GPLDA system under different duration conditions in the NIST SRE 2010 dataset..... 4

Table 3.1 Performance (EER%) of speaker verification sub-systems and lexical content sub-systems with different states and segments on Part 1 of RedDots database (male part only)..... 48

Table 3.2 Performance (EER%) of mixture selection with various mixtures on Part 1 of RedDots (male part)..... 49

Table 4.1 Performance (EER%) using i-vectors and N-vectors on SRE'10 8CONV-CORE, 8CONV-10SEC, 8CONV-5SEC and 8CONV-3SEC conditions 53

Table 4.2 Trace $\bar{\sigma}$ of the covariance matrix of supervectors in the i-vector framework..... 53

Table 4.3 Phonetic group information 70

Table 4.4 Performances (EER% and MinDCF%) of standard TVM, and TVM with MoG prior systems on the NIST SRE'10 CORE-EXT, CORE-CORE, and CORE-10sec sets with CC5 conditions..... 70

Table 4.5 Performance (EER%) of the baseline system, proposed and fusion systems on the NIST SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions 72

Table 4.6 Performance (EER%) of the baseline system, proposed and fusion systems on the NIST SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions 74

Table 5.1 EER results of speaker-phonetic vector representation on the male parts of the NIST SRE 2010 database 93

Table 5.2 MinDCF results of speaker-phonetic vector representation on the male parts of the NIST SRE 2010 database..... 94

Table 5.3 EER Results of speaker-phonetic vector representation on NIST SRE 2010 of female part 94

Table 5.4 MinDCF results of speaker-phonetic vector representation on NIST SRE 2010 of female part 95

Table 6.1 Comparison of the performance (EER%) of the standard baseline proposed methods evaluated on the NIST SRE'10 8CONV-10SEC condition as well as the additional 5 seconds and 3 seconds conditions 111

Table 6.2 Comparison of the performance (EER% and MinDCF%) of the standard and proposed methods evaluated on the NIST SRE'10 8CONV-10SEC condition as well as additional 5s and 3s conditions $\alpha = 0.75, \beta = 0.25$..... 115

Table 6.3 Comparison of the performance (in terms of EER % and MinDCF %) of standard , proposed method evaluated on the NIST SRE'10 8CONV-10SEC condition as well as additional 5s and 3s conditions $\alpha = 1.0, \beta = 0.0$..... 116

Table 7.1 Performance (EER%) using the standard and the proposed twin model GPLDA systems on the NIST SRE '10 8CONV-10SEC and additional 5SEC and 3SEC conditions (male speakers) 135

Table 7.2 Performance (EER%) using the standard and the proposed twin model GPLDA systems on the NIST SRE '10 8CONV-10SEC and additional 5SEC and 3SEC conditions (female speakers) 136

Table 7.3 Performance (EER%) of TM-GPLDA with uncertainty propagation on SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions (female speakers only) with different values of scaling factors (λ_L, λ_S) with 500 speakers in training data 139

Table 7.4 Performance (EER%) of GPLDA, GPLDA with uncertainty propagation (GPLDA_UP), TM-GPLDA and TM-GPLDA with uncertainty propagation (TM-GPLDA_UP) on SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions, with different numbe 139

Table 7.5 Performance (MinDCF%) of GPLDA, GPLDA with uncertainty propagation (GPLDA_UP), TM-GPLDA and TM-GPLDA with uncertainty propagation (TM-GPLDA_UP) on SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions, with different number 140

1 INTRODUCTION

1.1 Research Overview

The problem of authentication and identification is very challenging and has a long history [1]. Authentication refers to the problem of confirming or denying a person's claimed identity. Approaches to the problem of authentication conventionally include (i) a person's possession ("*something that you possess*"), e.g., keys; (ii) person's knowledge of a piece of information ("*something that you know*"), e.g., PIN. More recently, biometrics is an alternative approach to the problem of authentication and identification.

Biometrics, which refers to the automatic identification of a person based on his or her physiological or behavioural characteristics [1], has drawn researchers' attentions for a long time. A biometrics relies on the uniqueness of this human characteristic, serving as a personal authentication tool for applications like access control. Biometrics then provides a way to establish an identity based on "*who you are*", rather than "*what you possess*" or "*what you know*". The relevance of biometrics in modern society has been reinforced by the need for large-scale identity management systems [2]. Since biometrics offers certain advantages such as negative recognition and non-repudiation that cannot be provided by tokens and passwords, biometrics has gained increased visibility and significance in modern society [2]. Biometrics such as fingerprints and irises etc., have been developed and implemented in the real world, facilitating social activities such as commercial transactions.

The human voice contains speaker identity information. It is common for human to identify a speaker's identity just by listening to his/her voice. For example, in phone calls, people usually first started to identify the speaker by voice. It has recently been found that there are special auditory cortices that is dedicated to audio processing and may contribute to the ability of

human to identify speakers by voice [3]. This supports humans' ability to recognise other humans' voices [4] and suggest that each individual has unique characteristics traits in his or her voice [5]. This claim is supported by research from many fields, including neuroscience [3, 4, 6], cognitive science [3] and machine learning [5, 7, 8].

Since human voice combines physiological and behavioural characteristics which is unique to a speaker, it is an attractive biometric that is non-invasive. This means that, unlike other biometric such as fingerprint, there is no need for physical contact. Human voice propagates through the air, can be recorded by microphone and then transmitted through different channels. This is an attractive solution for many scenarios. For example, in telephone bank, authentications are frequently required in the process of transaction. Voice based biometric can be adopted to provide banks with an efficient tool to perform this authentication. In many other applications such as access control, voice-based biometrics is user friendly and easy to use as speaking is almost effortless to anyone.

Moreover, since the intelligent virtual personal assistants are popular, voice based biometric has become even more attractive. The intelligent virtual personal assistants are expected to communicate with human smoothly. As speech is the main medium of communication for human, these intelligent virtual personal assistants should be able to understand spoken demands and execute only orders given by the right person. The commercialised intelligent virtual personal assistants such as Siri from Apple Inc [9], smart home management systems such as Google Home [10], and smart loud speaker such as Alex from Amazon [11] can execute human's orders if given the right commands. However, some information may be only accessed by particular persons, such as parents in a family and some commands should only be executed by the order of the owner. In those cases, voice-based biometrics can be used to secure transactions, information, and premises to authorized individuals.

Voice based biometrics consists of automatic speaker identification and verification and the later one is often referred to automatic speaker verification (ASV). Automatic speaker identification refers to technology that allows machines to identify a speaker's identity from given samples of their voice. Automatic speaker verification then refers to technology that enables machines to verify a person's identity using their voice samples. ASV has gained more attention since it has wider applications and can be broadly categorised into one of two types, namely, text-dependent (TD) and text-independent (TI) systems. In TD systems, the contents of pass-phrases are fixed, providing extra information to verify the identity of the speaker [12]. In the TI case, speakers are free to speak any phrases and the system cannot rely on prior knowledge of fixed pass-phrases [8].

1.2 Short duration speaker verifications

A number of public organizations and academic organizations are interested in developing automatic speaker verification. For example, the National Institute of Standards and Technology (NIST) hosts the Speaker Recognition Evaluation (SRE) challenges and releases databases which help to advance speaker recognition techniques. The aim of speaker recognition is to determine whether a specified target speaker is speaking during a given segment of speech [13] and the protocols are compatible with speaker verification. During the last few decades, researchers in this area have had many breakthroughs and significant improvements of performance have been obtained. More than an eight-fold improvement has been observed within ten years [14].

Nevertheless, automatic speaker verification is not a solved problem. Most of the challenge databases [13, 15] including NIST SRE are recorded under ideal conditions, i.e. clear, open places, and each utterance has a long duration. For example, each utterance is 2.5 minutes long in the core conditions in NIST SRE challenges [13]. Currently, research focuses on problems

under more ‘wild’ conditions. For example, the wild speaker recognition challenge [15] and NIST SRE 2012 [16] focus particularly on noisy conditions. Moreover, in scenarios like access control, the long duration requirement of an utterance may not valid or applicable. Short duration (e.g., 5 seconds or less) is to be preferred. For example, in the RSR 2015 database, the average duration of a command is 3.2 seconds [17]. Reduced accuracy with short duration utterances is one of the most challenging problems for wider application of speaker verification technology.

However, even the performance of a state-of-the-art i-vector/GPLDA ASV system degrades sharply with decrease in duration of speech [18] as shown in Table 1.1 and Figure 1.1.

Table 1.1 Performance (EER %) of the i-vector/GPLDA system under different duration conditions in the NIST SRE 2010 dataset

Condition	EER (%)
8CONV-CORE	1.51
8CONV-10SEC	5.03
8CONV-5SEC	10.73
8CONV-3SEC	17.68

The i-vector/GPLDA system is used [19, 20] and it is tested on the NIST SRE 2010 database. Test data are truncated into 5 and 3 second segments to create the additional 8CONV-5SEC and -3SEC conditions. 8CONV stands for eight conversions, which means the enrolment data contains eight long utterances with around 2.5 minute long. Equal error rate (EER), which will be introduced in Section 2.6, is used as the measure of performance in this table and figure. From this table and figure, it is observed that performance of short duration condition degrades sharply. To tackle this problem, short duration conditions have also been re-introduced since NIST SRE 2008 [13], but it is not in core conditions. In the NIST SRE 2016 challenge [21],

durations of test utterances are randomly truncated into 10 seconds to 60 seconds and durations of some test utterance are even below 10 seconds.

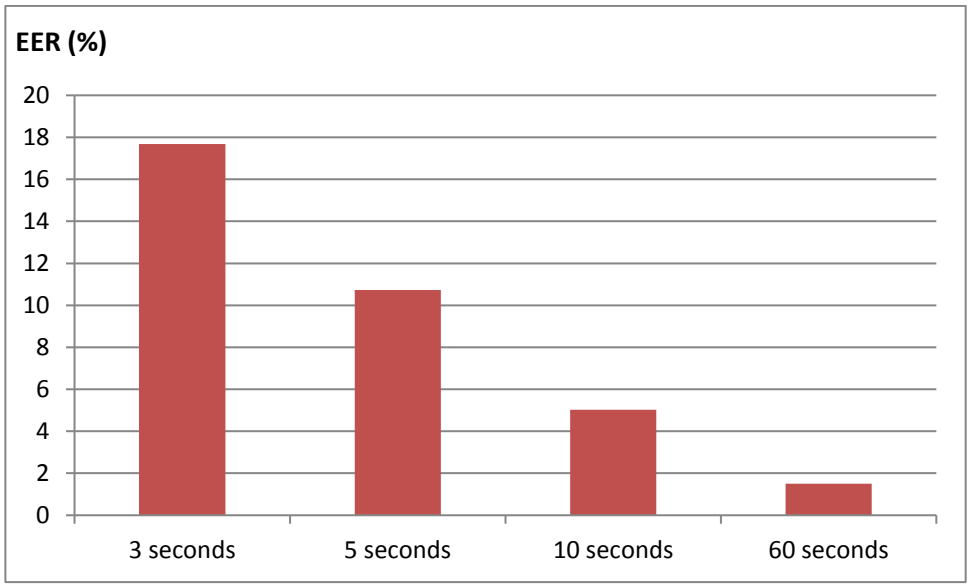


Figure 1.1 Speaker verification system performance over duration.

Moreover, as previously mentioned, humans are able to verify a speaker's identity effectively and with considerable ease without any conscious training if he or she is familiar with the identity who is speaking [5]. This suggests that a machine may also verify an individual's identity with relatively short duration utterances, given a large amount of speech used to familiarise the system with a particular individual. The research of short duration ASV will extend our knowledge of the human voice in terms of discrimination ability, and the degree to which a voice can represent one's identity.

1.3 Research targets

This thesis focuses on the problem of short duration ASV, which is termed short duration speaker verification throughout the rest of the thesis. Since enrolment is carried out once in an offline manner, it is therefore reasonable to assume long utterances are available for this. This is also consistent with how humans recognise individuals when they are very familiar with the

individual's voice. Similar to the NIST SRE challenges, utterances of less than 10 seconds in test files are defined in this thesis to be short utterances. Different durations, such as 5 seconds and 3 seconds, are also investigated and termed as extremely short utterances in this work.

Identity related discriminative information is distributed over time in speech. Short utterances necessarily mean that less information is accumulated. A profound question to ask is how to effectively encode short utterances so that machines can recognise an identity based on speech more accurately.

Another issue that arises in short duration speaker verification is the problem of duration mismatch. This occurs because of the fact that the amount of information is varying from one utterance to another. If the amount of information in different utterances is radically diverse, it is likely that some of the assumptions of speaker verification systems are violated, causing further problems. Thus, duration mismatch also needs to be considered.

1.4 Thesis structure

This thesis focuses on developing novel modelling and compensation techniques for short duration speaker verification. Chapter 3 proposes parallel speaker and content modelling techniques for text-dependent speaker verification. Chapters 4 and 5 attempt to generate better modelling techniques for short duration in ASV in the front-end, by proposing better vector representation of utterances and better models. Chapter 6 attempts to solve the problem of duration mismatch in utterance vector representation spaces by proposing suitable compensation techniques. Chapter 7 attempts to further compensate for such duration mismatch in the back-end by using two separate distributions to model long and short utterances. Specifically, this thesis has the following structure.

Chapter 2 reviews the basic principles of speaker verification technology. These include ASV system structure, speech production, feature representations, utterance representations, and different ASV systems. It then highlights the problem of short duration speaker verification with long duration enrolment and short test data.

Chapter 3 focuses on text-dependent (TD) speaker verification. A challenge faced by TD ASV is how to effectively model the alternative hypothesis as there are two tasks. In order to use both the content and speaker information in enrolment and test data, we propose the use of two separate sub-systems, based on hidden Markov models and sets of segment GMMs, in parallel to model the combined speaker and lexical content information in short duration utterances. In addition, the use of a mixture selection method is also proposed.

Chapter 4 begins with the deficiencies of i-vector representations for short utterances. i-vectors are a state-of-the-art technique utilised in long-duration speaker verification systems, but it is found they are not accurate for short durations in this chapter. The generalised variability model (GVM) is proposed for short duration utterances to generate more accurate representations, which is called i_g-vector, of utterances. It was also found that the information in each phonetic group, referred to as the local acoustic variability, is complementary to the total variability model. It then proposed a local acoustic model that utilises this information to complement the total variability model.

As alternatives for i-vector representation, Chapter 5 investigates more reliable representations for short utterances. First, it analyses the i-vector representation is sensitive to phonetic mismatches in short duration utterances. In order to relieve the problem, the speaker-phonetic vectors are proposed to have both speaker and phonetic meaning. Three models are proposed. The first one is the revised GVM to have speaker-phonetic vector representations. The second model is mixture of total variability model. The third one is the tied the parameters of mixture of total variability model.

After generating the vector representations in Chapter 4 and 5, it is still observed that duration mismatch appears in these vector spaces. Analysis in Chapter 6 indicates that vectors from long and short utterances are not sampled from the same distribution. Duration mismatch compensation techniques in the utterance vector representation spaces are then proposed to compensate this mismatch.

After compensation in the vector representation space, a back-end is applied to generate scores. From the analysis in Chapter 6, model vector representations with one single model may not be optimal. In Chapter 7, it is shown that duration mismatch will be propagated into back-end and will degrade performance. Two different distributions to model the vector representations of long and short utterances are then proposed. This is realised by tying the same latent variables for the two distributions. Following this, uncertainty propagation technique is incorporated into the model in order to take the uncertainty of latent variables in the previous vector representations.

Chapter 8 concludes the thesis and discusses future work.

1.5 List of contributions

The goal of this thesis is to improve the accuracy of speaker verification systems on test utterances that are less than ten seconds long. Novel modelling and compensation techniques for short duration speaker verification are proposed and analysed in this thesis. To conclude, this thesis:

I. finds contents and speaker discriminative information which are important for text-dependent short duration speaker verification, and proposes an improved parallel system to have a better model [Chapter 3];

II. proposes a generalized variability model to have better representation of utterance as an alternative for the i-vector representation [Chapter 4];

III. analyses the local acoustic variability of short duration utterances and proposes a complementary local acoustic variability model compared to the total variability model [Chapter 4];

IV. analyses phonetic i-vectors and finds that different phonemes in a utterance are not mapped to same distribution, which accounts for the phenomenon that short duration is sensitive to content mismatch [Chapter 4];

V. proposes three methods, to represent an utterance with vectors of both phonetic- and speaker–information, instead of representing one utterance with one vector representation of utterance, to cope with the content mismatch issue described above and described in Chapter 4;

VI. analyses the distribution of vector representations of short utterance and ascertains that they do not match the distributions corresponding to long utterances and then proposes duration compensation methods including Twin Model projection and deep neural network (DNN) based non-linear compensation methods to cope with this problem [Chapter 5];

VII. proposes a new back-end (Twin Model GPLDA) that is tailored for the scenario with long enrolment and short test files to cope with duration mismatch in vector representation spaces to generate more reliable scores. Uncertainty propagation is also integrated in this algorithm and described in Chapter 6.

1.6 List of publications

This thesis is based on the following publications, which are either published or submitted for publication:

Journal papers

- Jianbo Ma, Vidhyasaharan Sethu, Eliathamby Ambikairajah, Kong Aik Lee, "Duration compensation of i-vectors for short duration speaker verification," *Electronics Letters*, vol. 53, pp. 405-407, 2017. [from Chapter 5]

- Jianbo Ma, Vidhyasaharan Sethu, Eliathamby Ambikairajah, Kong Aik Lee, "Generalized variability model for speaker verification," *IEEE Signal Processing Letters* 25, no. 12 (2018): 1775-1779 [from Chapter 3]

Conference papers

- Jianbo Ma, Vidhyasaharan Sethu, Eliathamby Ambikairajah, Kong Aik Lee, "Twin Model G-PLDA for Duration Mismatch Compensation in Text-Independent Speaker Verification," *INTERSPEECH* 2016, pp. 1852-1857, 2016. [included in Chapter 6]
- Jianbo Ma, Saad Irtza, Kaavya Sriskandaraja, Vidhyasaharan Sethu, Eliathamby Ambikairajah, "Parallel Speaker and Content Modelling for Text-dependent Speaker Verification," *INTERSPEECH* 2016, pp.435-439. [from Chapter 3]
- Jianbo Ma, Vidhyasaharan Sethu, Eliathamby Ambikairajah, Kong Aik Lee, "Incorporating Local Acoustic Variability Information into Short Duration Speaker Verification," Proc. *INTERSPEECH* 2017 (2017): 1502-1506. [included in Chapter 3]
- Lee, K A. Hautamäki, V. Kinnunen, T. Larcher, A. Zhang, C. Nautsch, A. Stafylakis, T. Liu, G. Rouvier, M. Rao, W. Alegre, F. Ma, J. Mak, M W. Sarkar, A K. Delgado, H. Saeidi, R. Aronowitz, H. Sizov, A. Sun, H. Nguyen, T H. Sahidullah, Md. Vestman, V. Halonen, M. Kanervisto, A. et al.. (2017). The I4U Mega Fusion and Collaboration for NIST Speaker Recognition Evaluation 2016. 1328-1332. 10.21437/ *INTERSPEECH*.2017-203. [from Chapter 5]
- Jianbo Ma, Vidhyasaharan Sethu, Eliathamby Ambikairajah, Kong Aik Lee, "Speaker-Phonetic Vector Estimation for Short Duration Speaker Verification," In *International Conference on Acoustics, Speech, and Signal Processing* (ICASSP) 2018. [from Chapter 4]

2 LITERATURE REVIEW

Human can differentiate between individuals' voices fairly well and this ability has drawn the attention of researchers [22, 23] even before the creation of computers. The early stages of research focussed on what are the elements of voice that sounds natural, like a human's voice. These interests led to the fundamental questions of ASV. These are how speaker discriminative information is conveyed through speech, and how human beings recognise the identity of the speaker of a segment of speech [24].

Researchers have tried to answer those fundamental questions for a long time. The first attempt was the notion of a voiceprint [25]. A voiceprint is formed from the spectrogram of a voice and is assumed to be unique to an individual. Even though this notion has been criticised, the spectrogram is still used in current speaker recognition systems. Instead of regarding speaker discriminative information as purely stemming from the spectrogram, modern speaker verification systems operate on the assumption that speaker discriminative information is a combination of results of anatomical differences inherent in the vocal tract, and in the learned speaking habits of different individuals [26]. All discriminative features, including those from a physiological aspect, like vocal tract, larynx, lungs etc., which are reflected in the spectrogram of speech, and psychological traits like speaking rate and choice of words can be used to recognise identity. The second question finds its solution in neuroscience. The latest findings are that Temporal Voice Areas (TVAs) exist in the human auditory cortex [27] that is dedicated to speech analysis. These and future findings will continue to facilitate the research of ASV.

In automatic speaker verification systems, raw speech signals are used to extract speaker discriminative information and then different models are built. A description of diagram of a

basic ASV system is introduced in Section 2.1. Feature extraction is introduced in Section 2.2. In Section 2.3, different back-ends of automatic speaker verification systems are reviewed. In Section 2.4, the development of short duration speaker verification is reviewed. Relevant databases and evaluation metrics are presented in Section 2.5 and Section 2.6 respectively.

2.1 Automatic speaker verification system-overview

Figure 2.1 shows a diagram of an automatic speaker verification system. In the ASV system, speakers need to enrol in a system and the data for this purpose is termed enrolment data. In the verification stage, the data needed is called test data. The data needed to train the parameters of models is called background data. In general, an automatic speaker verification system consists of two phases. The first phase is to enrol target speaker in the system. Enrolment data will be used to train models and is completed offline. Two different sets of models are created in this phase. The first type is the background model, termed the universal background model (UBM), trained on speech from many speakers from background data. This is to develop an overall distribution of speaker features. Different features are extracted at this stage and details of feature extraction will be discussed in Section 2.2. Gender dependent UBMs are usually created. The second group of models is that of speaker models. As the enrolment data for each speaker model is limited, the UBMs are adapted using enrolment data to create speaker models. This process also removes mismatch between different speaker models.

The second stage of speaker verification is the verification phase. The ultimate goal of automatic speaker verification system is to generate scores for each claim. A decision is then made based on these scores to either reject or accept the tested claim depending on whether or not the score surpasses the system threshold. The lower panel of Figure 2.1 illustrates this process. Both the claimed speaker model and UBM are used by the pattern matching algorithm

to produce the score before entering the decision maker. Usually, the procedure to have vector representations of utterances is called front-end and the procedure to model these vector representations and to generate verification scores is called back-end. Techniques used in this diagram will be reviewed in the next few sections.

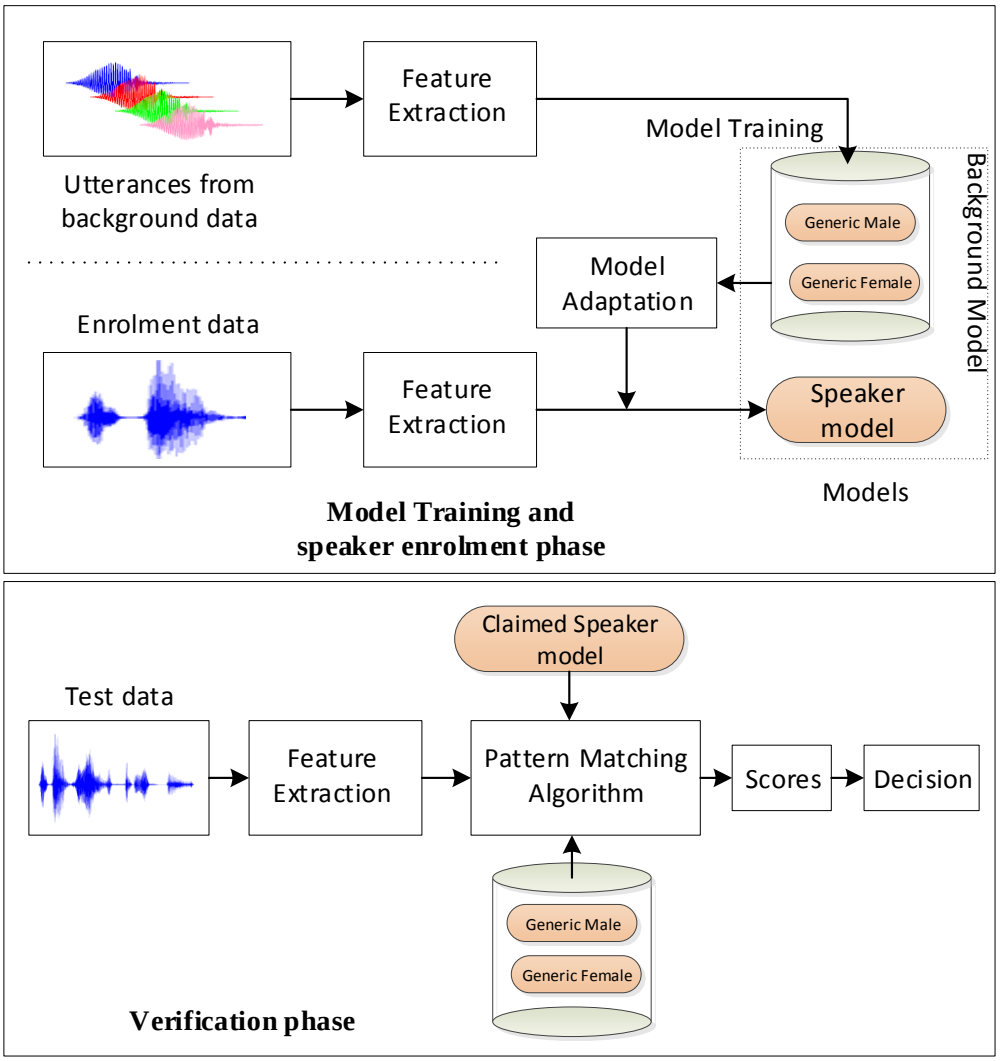


Figure 2.1 Diagram of a basic automatic speaker verification system, showing phases of model creation and speaker enrolment(top) and verification (bottom).

2.2 Front-end of automatic speaker verification system

The front-end of ASV system is to extract useful information from raw speech waveform and represent the speech files with a stream of vectors. This vector representation is called feature in ASV system. For speaker recognition tasks, an ideal feature for speaker verification would 1) have large inter-speaker variability and small intra-speaker variability; 2) be robust against noise and distortion; 3) occur frequently and naturally in speech; 3) be difficult to impersonate/mimic; and 4) not be affected by the speaker's health or long-term variations in voice [8, 28, 29].

As mentioned before, speaker discriminative information is originated from physiological and behavioural traits of speakers. This information can be represented in different perspectives such as short-term spectral, prosodic, etc. and a high level perspective which captures behavioural meaning [8, 14]. Figure 2.2 presents the discriminative information found in these different levels of speech. At the bottom, are the physiological aspects of speech. These features include short-term spectral, energy patterns, reflecting size of the vocal tract folds, length and dimensions of the vocal tract. Features at this level are easy to extract and less data is needed. But they are also less tolerant to noise and distortions and can easily lead to mismatches. In the middle are prosodic features, including speaking rate, durations etc. Features at this level reflect more about the psychological aspects of speakers and like the lower level physiological features, they are not text- and language dependent. Features at the top level of the diagram are high level features such as semantics, accent, lexical idiolect etc. They are learned behavioural features, which reflect socio-economic status, education, etc [8]. High level features attempt to use phoneme or world level characteristics of speakers. For example, the habits of different speakers in choosing particular words during conversation also have discriminative power. In [30], the author extracted N-gram frequency features which is one of high level features in conversations and showed that they can be used in verification systems. In [31], several kernels were proposed to transform the N-gram phoneme frequency in utterance and significantly reduced error rates

were observed when compared with conventional phoneme techniques. More recently, [32] used a statistical model, the Gaussian mixture model (GMM), to determine the parameters of the prosodic feature distribution, yielding a more generative model for prosodic features. Additionally, instead of using phonemes or words as a tokenizer, the authors of [33] used a GMM as a tokenizer to extract high level information. This is more computationally efficient. Although high level features require more data to extract a reliable feature representation of a speaker, they are shown to be more robust against noise and channel variability. The channel variability occurs as the speech is recorded under different environments or different devices. In fact, humans are very efficient at extracting high level features and can take advantage of these features.

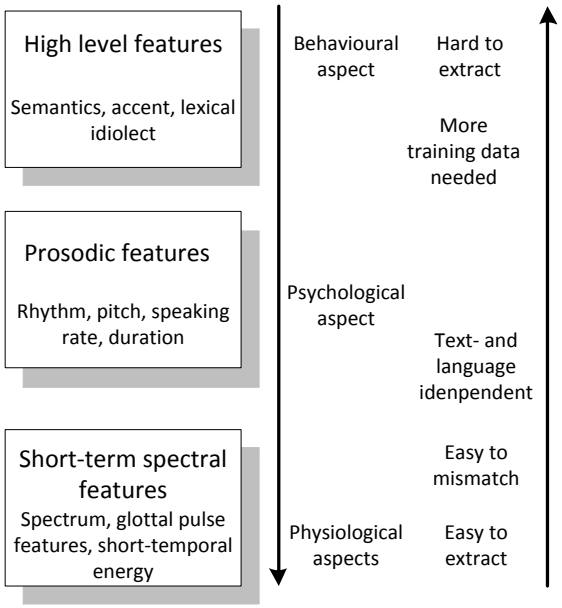


Figure 2.2 The different levels of speaker discriminative features [8].

Recently, DNN based features have also been introduced into automatic speaker verification systems. Among these, bottleneck features [34, 35] have been applied frequently. Bottleneck features are derived from a layer that is several layers away from the last layer of a deep neural

network. At this layer, the dimension is small (e.g., 42 [35]) and is called a bottleneck layer. This layer is conventionally trained from an automatic speech recognition system and is expected to capture phonetic information. Other DNN based features are trained to optimise an objective function to distinguish between speakers [36].

2.2.1 Feature extraction

Of the different levels of features available shown in Figure 2.3, short-term spectral features are the most discriminative and simplest features in the literature, and they are commonly used in automatic speaker verification systems [8]. Among these features, Mel-frequency cepstral coefficients (MFCCs), linear frequency cepstral coefficients (LFCCs), linear prediction cepstral coefficients (LPCCs), perceptual linear prediction cepstral coefficients (PLPCCs) are the most commonly applied features [37, 38] in speech and audio processing.

Figure 2.3 presents the procedure of MFCC feature extraction. The aim of this procedure is to obtain vector representations of a segment of speech signal so high dimensions in time domain can be avoided. The input segment of speech is first framed with frame length (e.g., 128). Discrete Fourier Transform like Fast Fourier Transform (FFT) can be applied on each frame to obtain the frequency response. A filter bank is then designed with the Mel frequency scale that simulates the frequency response of a human ear [37]. In the Mel-scale, the scale of frequencies is not a linear scale, with more filters assigned to the low frequency region. The signal energy is extracted from these filters and the logarithm function is applied to the energy values in different frequency bands, resulting in a series of energy coefficients. The discrete cosine transform (DCT) is used to de-correlate these coefficients [39]. Finally, the coefficients are concatenated with their temporal derivatives, and voice activity detection (VAD) is applied to remove unvoiced parts of the speech signal. A segment of speech will be represented by a number of successive frames of MFCCs.

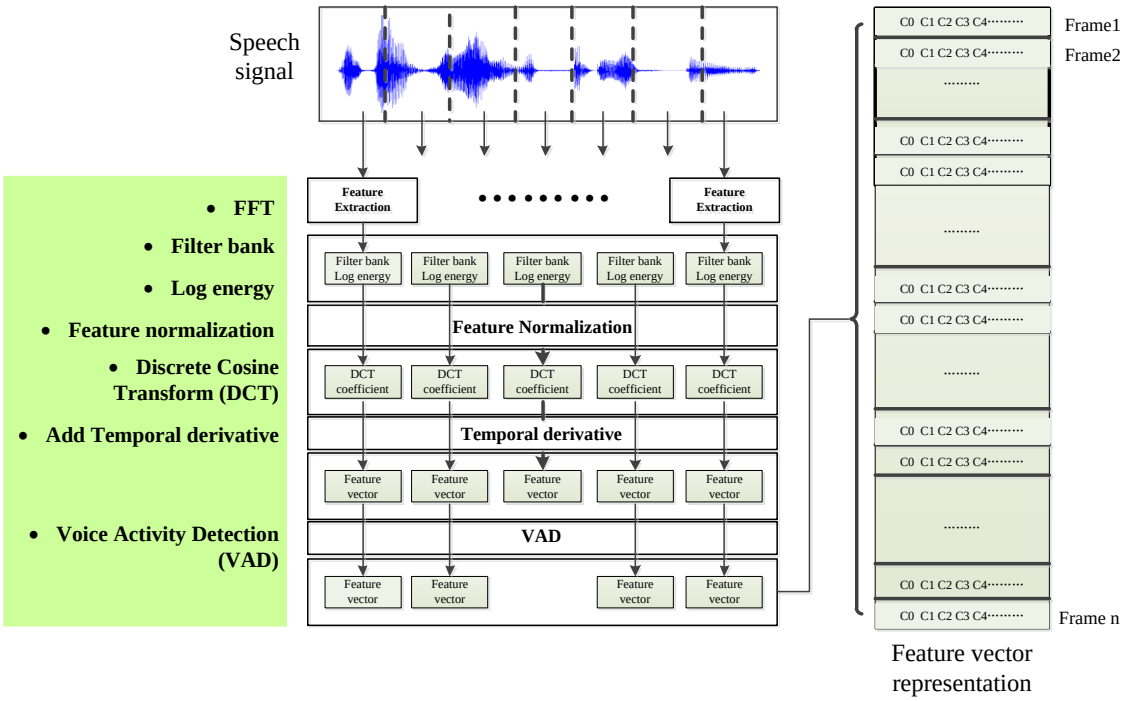


Figure 2.3 Extraction process of Mel-frequency cepstral coefficients (MFCCs).

Other short-term spectral features have similar extraction procedures, with the difference that uses different scales, such as Bark scale [40], or using different models to model time series, like perceptual linear predictive cepstral coefficients (PLPCCs) [41].

2.2.2 Feature normalisation

Before feeding features into the modelling or matching algorithms, the features should be normalized because they come from different microphones and different environmental noise conditions [42]. Every microphone is different, and their sensitivity will affect the value of features when they record speech. Different channels will also cause different types of artefacts to appear on a recorded signal, and these differences will cause mismatch in the recognition algorithm. Normalisation of the features to suppress added artefacts caused by different channels is needed. In [43], it is shown that slow changing channels appear as an offset to the individual coefficients of the cepstral vector. This is because of the fact that nonlinear (e.g., convolutive) channel effects become linear and additive in the log-spectral and cepstral

domains. Cepstral Mean Subtraction (CMS) [43] uses this fact and subtracts a constant value over a certain number of frames. For example, the constant values can be the mean vector over the whole utterance or a shorter period of time, on the order of 3-5s. By subtracting this constant value, features obtained from different channels all become zero-mean, leaving only the relative relationships between frames. By this method, the channel effect is significantly reduced. Similar to CMS, relative spectral (RASTA) filters [44] use a moving average filter in the log-spectral or cepstral domain. Feature warping (FW) [45] is another effective feature normalisation method. The distribution of speech is typically not Gaussian and varies with different utterances. This fact will affect the accuracy of a model. FW transforms the differently distributed features into a uniform distribution, called the target distribution, by warping the cumulative distribution function of the features to match the target distribution. Generally, the values of features are generated from a lookup table using the relative positions of the features during a short window, e.g., 3 seconds. Other normalization methods like feature mapping (FM) [46] are known as supervised normalization methods. First, the most likely channel is detected and then the features are mapped into a channel-independent space using the recognized channel model. Different channels have different mapping functions. In the channel-independent space, features are assumed to be channel invariant.

2.3 Back-end of automatic speaker verification systems

After feature extraction, different models are built on top of the different feature sets to model speakers, and back-ends are used to compare speakers.

2.3.1 GMM-UBM system

As mentioned in previous sections, in ASV, a classifier is used to determine whether a stream of features originates from the claimed speaker. Using the features directly is not realistic because

of the high computational and memory requirements. Instead of using the stream of features directly, speaker models should be built before being fed into a classifier.

The first method of speaker models is to compress the training features to a set of vectors. That is the main idea of the Vector Quantization (VQ) [47] codebook model. Basically, given a set of training features, VQ will find a partitioning of the feature vector space for that particular speaker where the whole feature space is represented by those partitions. Each partition forms a convex, non-overlapping region and every vector inside the partition is represented by the corresponding centroid vector. The partition can be performed using clustering algorithms like K-means [48] to minimize the distortion over the whole training set. This distortion can be measured by Euclidean distance. In the verification phase, the average quantization distortion is used as the criteria to give a decision. This method is efficient when modelling a speaker compared with the stream of training data, but since it makes hard decision for a vector partition, the accuracy is limited and performances are unsatisfactory. Statistical methods need to be proposed to model speakers.

Instead of using hard decision, a GMM [49] is a stochastic model that has become the main-stream modelling method in ASV. Several Gaussian components can be used to model the distribution of feature vectors. One advantage of GMM is that a linear combination of Gaussian basis functions is capable of representing any arbitrary distribution. In GMM-UBM system, the UBM is a trained GMM and is a model of the distribution over the feature space for any speech. Each Gaussian component in UBM is assumed to cover a group of closely related acoustic classes. These classes may represent some broad phonetic events, such as vowels, nasals, or fricatives, and reflect some general speaker-dependent vocal tract configurations that are useful for characterizing speaker identity [49]. Each component of the GMM has three parameters: weight, mean and covariance. The weight can represent the prior probability of a particular class. The mean can represent the spectral shape of the acoustic class and the covariance

represents the variations of the average spectral shape [49]. To obtain the parameters of a GMM, the expectation maximization (EM) algorithm [50] in Appendix A is used to iteratively estimate the parameters.

Suppose the observed data as $x = \{x_1, x_2, \dots, x_N\}$. The Gaussian mixture distribution of this data with K mixtures is represented as

$$p(x_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \boldsymbol{\Sigma_k}) \tag{2-1}$$

where π_k is k^{th} mixture's mixing coefficient, μ_k and Σ_k are the k^{th} mixture's mean vector and covariance matrix, respectively. $\mathcal{N}(x_n|\mu_k, \boldsymbol{\Sigma_k})$ is the Gaussian distribution, which can be written as

$$\mathcal{N}(x_n|\mu_k, \boldsymbol{\Sigma_k}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma_k}|^{1/2}} exp\left\{-\frac{1}{2}(x_n - \mu_k)(\boldsymbol{\Sigma_k})^{-1}(x_n - \mu_k)^T\right\} \tag{2-2}$$

where D is the dimension of the observed data. Parameters of GMM includes $\theta = \{\pi_k, \mu_k, \Sigma_k, k = (0,1, \dots, C)\}$, where C is the number of mixture component. Those parameters can be trained by the EM algorithm and details are in Appendix B.

However, it is not efficient to build one GMM for each speaker as there is not enough data. A conventional GMM-UBM structure and it is illustrated as Figure 2.4. Each speaker may also have different distributions, making it difficult to compare different speakers. Instead of one GMM per speaker, a speaker-independent universal background model (UBM) will first be built [51]. The UBM is a large GMM trained to represent the speaker-independent distribution of features using as many background speakers' features as possible. Those background speakers are from outside of the list of speakers for authentication. The speaker's model is adapted from

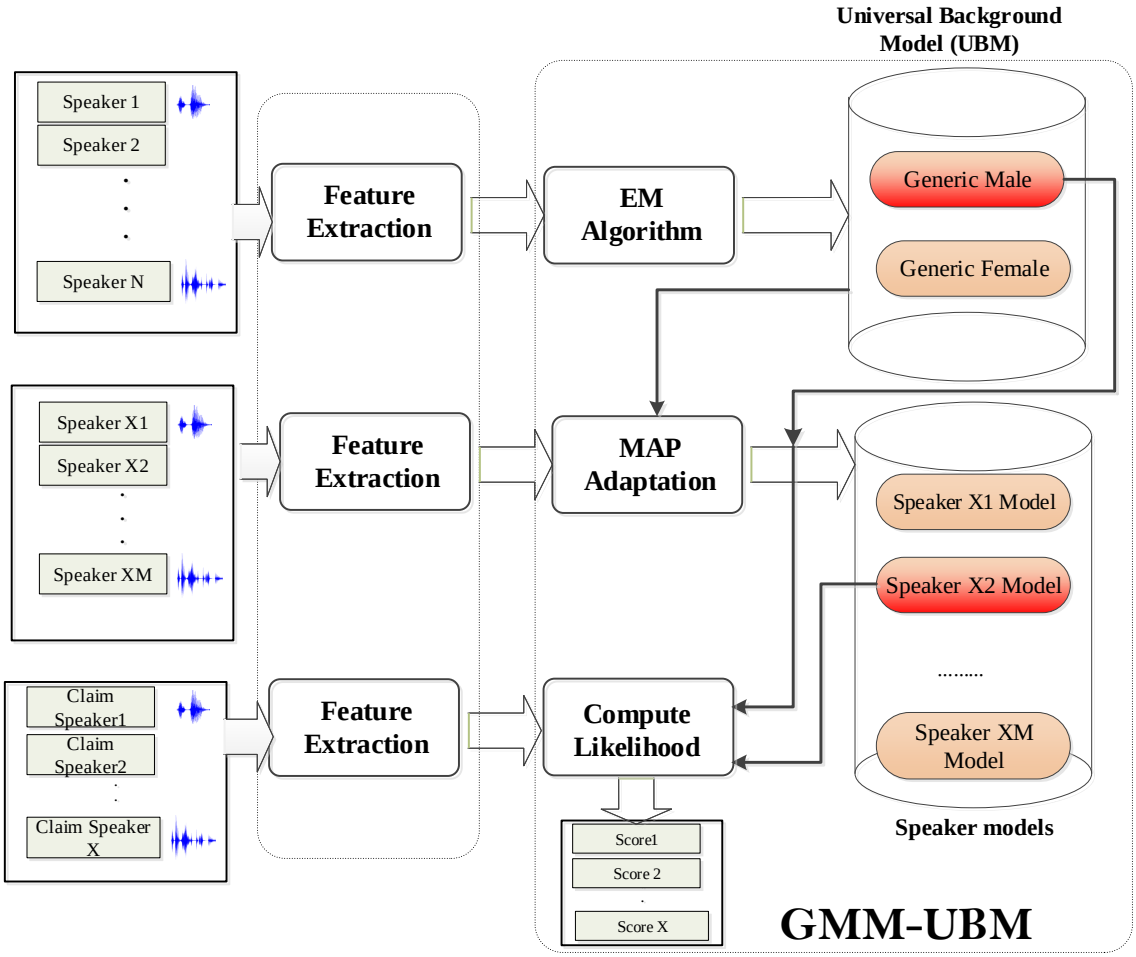


Figure 2.4 A diagram of a conventional GMM-UBM system [49].

the UBM by using the Maximum a Posteriori (MAP) adaptation [51]. The updated parameter formulas for MAP adaptation are

$$\hat{\pi}_k = \frac{\sum_{n=1}^{N_k} r(z_{nk})}{\sum_{n=1}^{N_k} r(z_{nk}) + \tau} \frac{\sum_{n=1}^{N_k} r(z_{nk})}{N_k} + \frac{\tau}{\sum_{n=1}^{N_k} r(z_{nk}) + \tau} \pi_k, \tag{2-3}$$

$$\hat{\mu}_k = \frac{\sum_{n=1}^{N_k} r(z_{nk})}{\sum_{n=1}^{N_k} r(z_{nk}) + \tau} E[x] + \frac{\tau}{\sum_{n=1}^{N_k} r(z_{nk}) + \tau} \mu_k, \tag{2-4}$$

$$\hat{\Sigma}_k = \frac{\sum_{n=1}^{N_k} r(z_{nk})}{\sum_{n=1}^{N_k} r(z_{nk}) + \tau} E[x^2] + \frac{\tau}{\sum_{n=1}^{N_k} r(z_{nk}) + \tau} (\Sigma_k + \mu_k \mu_k^*) - \hat{\mu}_k{}^2. \tag{2-5}$$

where z is a K-dimensional binary random variable and the definition can be found in Appendix B, τ is a relevance factor that controls the adaptation, whether data-dependent or mixture-dependent; γ is a scale factor that ensures the adapted mixing coefficients $\hat{\pi}_k$ sum to one. Then every speaker model has the same configuration and is more accurate to model a speaker's distribution. In the recognition phase, test feature frames are aligned to both a claimed speaker model and the UBM. Average log-likelihoods from the speaker and UBM model are computed and used to form the final score. The higher the score, the more likely the test utterance is from the speaker model. Otherwise, it more likely comes from the background speakers (UBM). The final decision will be determined according to a set threshold and the final score [51].

2.3.2 Supervector with Support vector machines

Support vector machines (SVM) were also introduced into speaker recognition [52-54]. Basically, the SVM uses a kernel function to map the low dimensional features into a higher dimensional feature space. In this high dimensional space, the mapped features from speakers and background speakers are assumed to be more discriminative. As shown in Figure 2.5, SVM defines a hyperplane which is used for separating two classes. This hyperplane and other parameters are trained and selected to optimize an objective function, so that the margin between two different classes is maximised. Support vectors are those vectors that are parallel and closest to the hyperplane. The distance between support vectors from the speaker model and background models is used as the score for decision making. The input feature to the classifier is now not the raw feature. Instead, supervectors derived by concatenating the mean vectors of the adapted GMM are used as features. Because the support vectors are computed beforehand, the time to compute a score is far less than the log-likelihood calculation of a GMM-UBM system.

The notion of the supervector that is applied in ASV is very important. A supervector is a high and fixed dimensional representation of an utterance [8]. That means an utterance with

arbitrary duration can be converted into a fix dimensional vector. The supervector is formed by concatenating the means of mixture components in GMM, which means the supervector $M = [\mu_1, \mu_2, \dots, \mu_k]$. This opens the door for other compensation methods in the machine learning field. For example, Nuisance attribute projection (NAP) [52] is used to remove the undesired channel variability from the supervectors before SVM training. Additionally, within-class covariance normalization (WCCN) [55] can be applied to compensate the SVM supervector. With those compensation techniques, system performance is significantly improved.

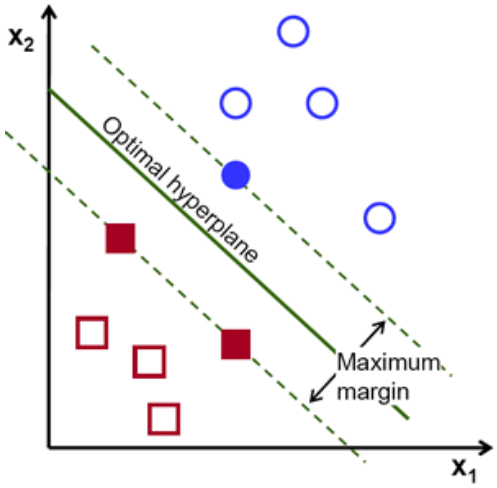


Figure 2.5 A diagram of Support Vector Machine process, showing the hyperplane separating two classes, and the closest support vectors that are used to define the hyperplane [56].

2.3.3 Total variability model with probabilistic linear analysis model

Following the supervector, the i-vector is another fixed dimensional representation of an utterance that has recently become popular [19]. The notion of i-vectors originates from joint factor analysis (JFA) [57] and once proposed, it soon became the state-of-the-art front-end of ASV. The first assumption of JFA is that the distribution of the supervector derived from the UBM is Gaussian and that it is speaker- and channel-dependent. The second assumption is that the supervector can be decomposed into a sum of two supervectors: one that describes the

speaker and another that describes the channel. Both of these are statistically independent and normally distributed. The third assumption is that the speaker supervector has a latent variable description in a much lower dimensional space (often two orders of magnitude smaller), which represents the speaker model and is called the speaker factor. The same is done for channel-dependent supervector and lead to channel factor. Based on the JFA technique, Dehak [19] found that the channel factor in the JFA model also contains speaker information and thus formed the total variability matrix and the notion of i-vector. The process of extracting i-vector from an utterance is depicted by Figure 2.6.

After the feature extraction phase, a UBM is built. In the TVM framework the probability density function of short-term spectral feature x of a speech signal is firstly modelled by this Gaussian mixture model (GMM) as (2-6). Instead of using k as the index of mixture component, c is applied and the range of it is $(1, C)$. For each utterance, an adapted GMM is created in order to model the distribution of the features of the given utterance. This GMM differ from the UBM only in terms of the means. The supervector is then assumed to represent the assigned GMM [58], though it has a large dimensionality (e.g., 40k). In order to represent the distribution more efficiently, the TVM maps the supervector into a lower dimensional space using the generative model which is specified by

$$M = M_0 + \boldsymbol{T}\omega \tag{2-6}$$

where M_0 is the supervector corresponding to the UBM, $\boldsymbol{T}$ is the factor loading matrix with rank R (e.g., 400) and termed as T matrix hereinafter. The means of the GMM are then mapped to a low-dimensional subspace ω.

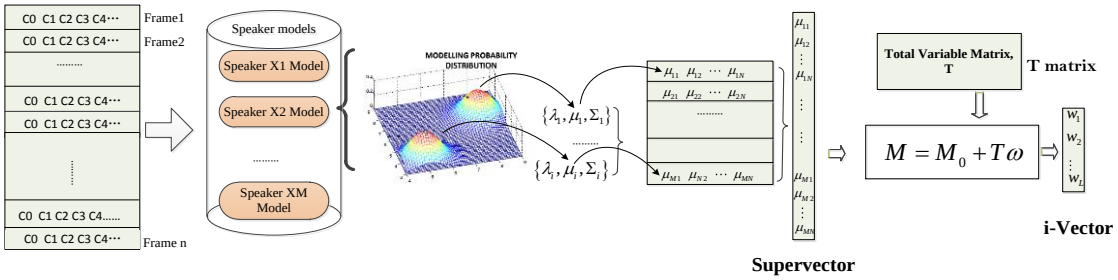


Figure 2.6 Diagram of the creation of a supervector from a set of GMM speaker models, and its conversion to the i-vector representation.

From (2-6), TVM is a factor analysis model and is shown in Figure 2.7. The variables z denote the labelling variables; x denotes feature frames; μ is mean of the supervectors; ω is the latent variable. The superscript i is utterance index; subscripts c denotes the mixture component in the UBM, and n is feature frame index. TVM is similar to probabilistic principal component analysis (PPCA) [59] in the aspect that latent variables are modelled as random variables drawn from a Gaussian distribution. The difference is that in TVM, supervectors are not directly used and are instead represented from Baum-Welch statistics estimated by a UBM. For a factor analysis model, a prior distribution over the latent variables is need. The prior distribution in TVM is a assumed to follow a standard normal Gaussian, which is,

$$p(\omega) = \mathcal{N}(0, \boldsymbol{I}) \tag{2-7}$$

where the bold $\boldsymbol{I}$ indicates an identity covariance matrix with the rank equal to the dimensionality of the mean vector. This means that the latent variables and supervectors are normal distributions.

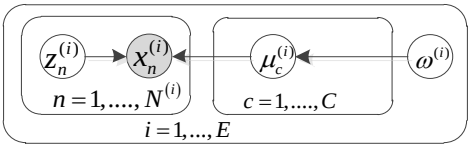


Figure 2.7 Graphical model representation of a total variability model. The variables are: z - labelling variables; x - feature frames; μ - means of the supervectors; ω - latent variable. The indices are: superscript i – utterance index; subscripts c - mixture component in UBM, and n - feature frame index.

Given the observed features $X = \{x_1, x_2, \dots, x_n, \dots, x_N\}$ extracted from an utterance, the posterior distribution of the latent variables is given as

$$p(\omega|X) \propto p(X|\omega)p(\omega). \tag{2-8}$$

The i-vector is the posterior mean of the latent variables ω given X and the model. It is calculated as

$$E(\omega|X) = \left(\boldsymbol{I} + \sum_{c} N_c \boldsymbol{T_c^*} \boldsymbol{\Sigma_c^{-1}} T_c\right)^{-1} \left(\sum_{c} \boldsymbol{T_c^*} \boldsymbol{\Sigma_c^{-1}} F_c\right) \tag{2-9}$$

where the $E(\cdot)$ is the expectation operator, T_c is the $F \times R$ (F is the dimension of feature x_n and R is the rank of T) dimensional sub-matrix of T corresponding to the c^{th} Gaussian mixture component of the UBM, N_c and F_c are the zero- and first-order statistics of the c^{th} component, respectively. zero- and first-order statistics are calculated as $N_c = \sum_n p(x_n|\theta_c)$, $F_c = \sum_n p(x_n|\theta_c)x_n$, where θ_c denotes the parameters of c^{th} component in UBM, $*$ is the transpose operator.

As the i-vector is in a low dimensional space, channel compensation methods like within-class covariance normalization (WCCN) [60], linear discriminant analysis (LDA) [61], and nuisance attribute projection (NAP) [62] can be effectively used. Kenny [20] introduced Probabilistic Linear Discriminant Analysis (PLDA) [63] as a back-end to the i-vector system and the i-vectors have become the de-facto technique to obtain fixed and low-dimensional representations of speech utterances for speaker verification [19]. However, in [20], a heavy tail t-student distribution is used to model the latent variable distribution. This models the non-Gaussian like behaviour of the i-vector. Compared to a Gaussian distribution, the t-student distribution is more computationally intensive. In order to solve this problem, [64] proposes to use length normalisation to make i-vector more normally distributed. Channel compensation

methods, like LDA, WCCN, and NAP [19] are also included in the pre-processing of the i-vector before scoring. The pre-processing can be described as

$$\widetilde{\omega} = \boldsymbol{W}\frac{\omega}{|\omega|} \tag{2-10}$$

where $|\cdot|$ is the $L2$ norm operator and $\boldsymbol{W}$ is a matrix that is a product of matrices from LDA, WCCN or NAP if those techniques are applied.

The standard Gaussian PLDA (GPLDA) is a generative model for i-vectors that has been successfully applied to deal with channel variability in speaker verification systems [20]. Given a set of pre-processed i-vectors $\chi = \{\widetilde{\omega}_{ij}\}$, where $i = 1,2,\cdots,S, j = 1,2\cdots,N^i$ (S is the number of speaker) and $\widetilde{\omega}_{ij}$ denotes the pre-processed i-vector corresponding to the j^{th} utterance from the i^{th} speaker, GPLDA decomposes them as:

$$\widetilde{\omega}_{ij} = \mu_g + \boldsymbol{\Phi} h_i + \varepsilon_{ij} \tag{2-11}$$

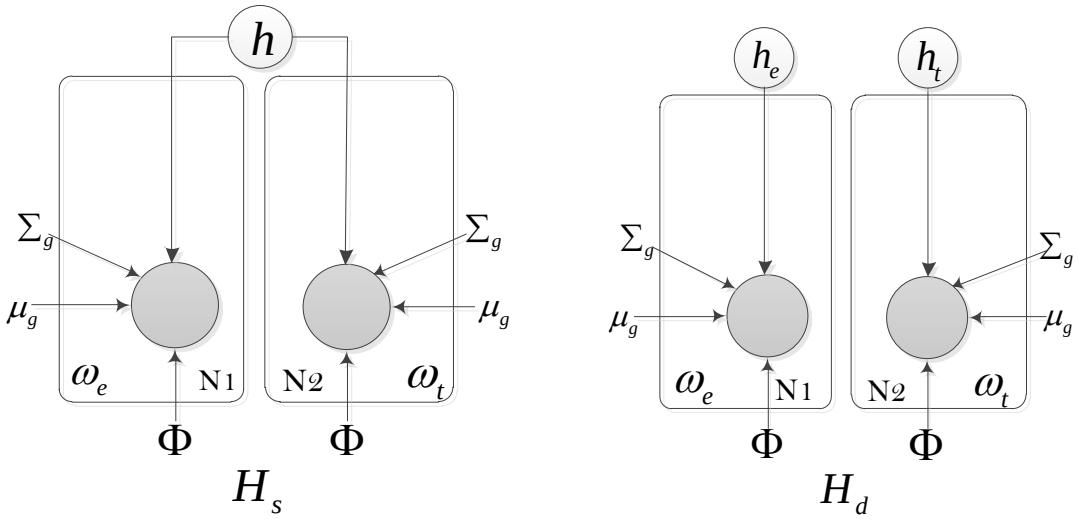
where μ_g is the mean of pre-processed i-vector, $\boldsymbol{\Phi}$ is a factor loading matrix, h_i is a vector of latent variables with a standard Gaussian distribution of $N(0, \boldsymbol{I})$, and ε_{ij} is a residual term that is assumed to be Gaussian with zero mean and a full covariance matrix denoted by Σ_g. The latent variables (elements of h_i) are assumed to be statistically independent. By marginalizing over the latent variables, the i-vectors follow a normal distribution given by $\mathcal{N}(\mu_g, \boldsymbol{\Phi\Phi}^* + \boldsymbol{\Sigma_g})$.

Based on this model, given an enrolment i-vector ω_e and a test i-vector ω_t from a trial, the hypothesis test is illustrated by the graphical representation in Figure 2.8. The log-likelihood ratio between the hypothesis that the two i-vectors are from the same speaker versus the hypothesis that they are from different speakers is calculated as follows [64]:

$$Score(x_e, x_t) = \frac{p(\omega_e, \omega_t|h)}{p(\omega_e|h_e)p(\omega_t|h_t)}$$

$$= \log\left(\mathcal{N}\left(\begin{bmatrix}\omega_e\\ \omega_t\end{bmatrix}; \begin{bmatrix}\mu_g\\ \mu_g\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Phi\Phi}^* + \boldsymbol{\Sigma_g} & \boldsymbol{\Phi\Phi}^*\\ \boldsymbol{\Phi\Phi}^* & \boldsymbol{\Phi\Phi}^* + \boldsymbol{\Sigma_g}\end{bmatrix}\right)\right) \quad (2\text{-}12)$$

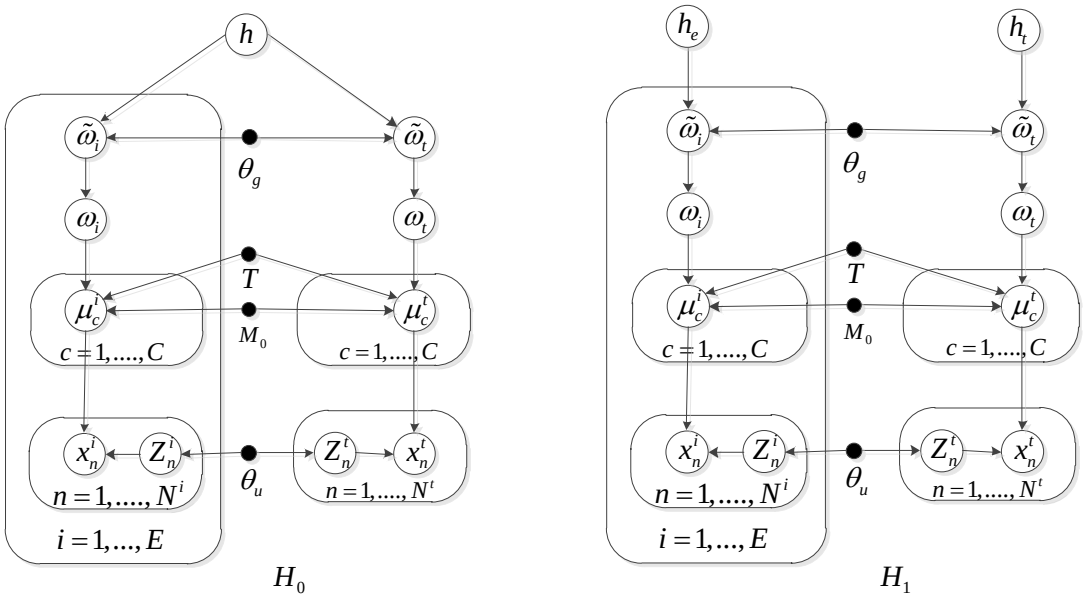
$$- \log\left(\mathcal{N}\left(\begin{bmatrix}\omega_e\\ \omega_t\end{bmatrix}; \begin{bmatrix}\mu_g\\ \mu_g\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Phi\Phi}^* + \boldsymbol{\Sigma_g} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Phi\Phi}^* + \boldsymbol{\Sigma_g}\end{bmatrix}\right)\right).$$



(a) (b)

Figure 2.8 The hypotheses that (a) the test and enrolment i-vectors are from same speaker (i.e., share latent variables h), and (b) that enrolment and test i-vectors are from different speakers (i.e. have distinct latent variables h_e and h_t).

Together with the i-vector model introduced in previous section, this forms the so-called i-vector/GPLDA system, which is represented by graphical model in Figure 2.9. Note that in this model, the non-linear transform of length normalization and channel compensation techniques have not been incorporated.



(a) (b)

Figure 2.9 Graphical model representation of i-vector/GPLDA system, where θ_g and θ_u *denote the parameters of GPLDA and UBM respectively,* E *is the total number of enrolment data, the super script* i *denotes parameter or variables are from enrolment data,* t *denotes parameter or variables are from test data. (a) Hypothesis that test and enrolment i-vectors are from same speaker (share latent variables - h); (b) Hypothesis that test and enrolment i-vectors are from different speakers (distinct latent variables -* h_e *and* h_t*).*

2.3.4 Neural network based models

Deep neural networks (DNNs) have been applied to automatic speech recognition (ASR) in recent years and has seen significant improvements [65]. In ASV, the utilisation of DNNs has been a fairly recent development. The first category of DNN application is the use of DNN functions to replace parts of the conventional system. The first notable application of a DNN in an ASV system replaced the typical UBM to provide zero- and first-order statistics [66, 67]. The structure of the ASR system remains the same as in Figure 2.8, with the main difference being that the GMM based UBM is replaced by the ASR system. In the ASR system, a DNN system is trained to classify an input to a number of phoneme or triphone classes which act similarly as mixture component in UBM. Conventionally, the statistics are calculated for observed feature

given mixture component in UBM. In DNN based i-vector/GPLDA system, statistics are calculated for observed feature given phoneme or triphone class. This replacement provides more accurate frame alignment and is able to improve verification performance in most scenarios. But in some cases, especially if the language of the target domain is different from that in the development domain, it does not provide much improvement or even worse results may be obtained [68]. Another application of DNN is in the front-end to extract different features. Bottleneck features are among this category and have also has been used with conventional short-term spectral features, providing complementary information [34, 35]. Other implementations include using a Boltzmann machine to train a discriminative classifier to generate scores[69, 70], making system more robust against noise [71].

The second category of DNN application is to have a different system structure compared with i-vector/GPLDA. Structures that are different with conventional UBMs system have been proposed. In [72], the use of an end-to-end DNN system is proposed that does not include several stages of training. The system is trained as a unified structure. The inputs of this system are the features and outputs are the final decisions. This system was proposed for text-dependent scenarios and was evaluated on the authors' own database, reporting better performance compared to a conventional i-vector system. Apart from this, there have been other implementations of DNNs in ASV as well. For example, [73] introduced the triplet loss [74] into the end-to-end structure to train an embedding of utterance representation for the speaker turn task which is to detect if speakers is changing in a segment. The name triplet loss arises from the fact that three samples that are positive, negative and anchor sample are included during training. The objective of the triplet loss is to reduce the distance (e.g., Euclidean distance) between positive and anchor sample, while increase distance between negative and anchor sample. Details about triplet loss can be found [74]. The embedding means a layer in DNN that embeds the information of input features. The implementation of [75] used the same triplet loss for embedding and showed good performance in short enrolment and short test

conditions. However, the current start-of-the-art i-vector/GPLDA system still has better results in long enrolment conditions [75]. Similarly, D. Snyder etc. trained a neural network system that mapped input features into corresponding speaker labels and a middle layer embedding that is extracted for the classifiers. It showed that it is beneficial to augment noisy data with clear training data during DNN training [76]. In this DNN implementation, the structure of TDNN [77] is used in the first several layers to accommodate wider temporal context and mean and standard deviation pooling layer is used to pool frame layer features and generate utterance level features. This utterance level feature is called x-vector and a subsequent G-PLDA is still used as a back-end. It shows the x-vector system improves the performances in most cases especially in noisy condition [76]. The use of DNNs in ASV are still active area of research.

2.4 Short duration speaker verification

In ASV, since the training phase is offline, it is easier to collect long utterances for speaker enrolment. However, in the test phase, short duration utterances are much more likely in many situations. For example, in access control type use cases the average utterance length is only 3.2 seconds [17]. However, different research endeavours have described different lengths for short utterances from around a minute to less than 10 seconds. The NIST SRE 2008 and 2010 database contains a condition for 10 second duration data. NIST SRE began to have 10 seconds data in 2008. Researchers started to show more interests in the short duration ASV. Recently, the meaning of a short utterance has been updated to mean less than 3 seconds of speech [17]. As such, short duration ASV does not have a very long history of research.

Short duration ASV can also be broadly divided into two classes, namely Text-Dependent and Text-Independent system, which has been mentioned in Section 1.1. TD ASV is not as challenging as the TI ASV, but is also quite applicable in real word. TD ASV systems have lexical constraints which require the speaker to speak specific pass-phrases, which are fixed

prior to authentication or prompted during the authentication process. TD systems are able to use shorter enrolment and test sessions than TI systems, so that TD systems are generally preferred for security authentication scenarios [78].

A recent study of TD speaker verification focuses on the efficient modelling of speaker and lexical content information of extremely short utterances (around 1.5 seconds) [17]. However, a challenge faced by text-dependent speaker verification systems is in framing the alternative hypothesis. The hypotheses in TI speaker verification systems are straightforward: the hypothesis under test, H_{χ}, denotes that test sentence is from the claimed speaker χ, while the alternative hypothesis $H_{\bar{\chi}}$ is that the test sentence does not come from the claimed speaker [51]. However, in TD speaker verification system, the hypotheses under test, $H_{(\chi,\mathcal{P})}$, is that the test utterance is from the claimed speaker and the content of the utterance matches the expected pass-phrase. Consequently, there are three potential alternative hypotheses, namely: $H_{(\bar{\chi},\mathcal{P})}$, that the speaker is not the claimed speaker but the pass-phrase is correct; $H_{(\chi,\bar{\mathcal{P}})}$, that the speaker is the claimed speaker but the test utterance is not the expected pass-phrase; and $H_{(\bar{\chi},\bar{\mathcal{P}})}$, that the speaker is not the claimed speaker and the test utterance is not the expected pass phrase [79]. These three alternate hypotheses may be referred to as imposter-correct, target-wrong and imposter-wrong hypotheses respectively.

The advantages of TD speaker verification over the TI ASV systems arise from having prior knowledge of the pass-phrase that is to be spoken which in turn allows for the use of more accurate content-specific speaker models. A recent approach to TD speaker verification has generalised the Joint Factor Analysis (JFA) mentioned in Section 2.4.3 framework to consider supervector-sized $\mathcal{Z}$-vectors that model speaker-phrase combinations with promising results [80]. More recently, $\mathcal{Y}$-vectors and $\mathcal{Z}$-vectors that are expected to characterize both passphrase and speaker information were jointly used to model the left-to-right temporal structure of utterances and a joint density back-end was proposed [81]. In terms of left-to-right structure, a

hidden Markov model (HMM) based system was also applied in [79]. Specifically, in [79], a hierarchical system including a GMM and HMM was proposed, and showed the benefits of modelling the alternative hypothesis. Besides this, DNN based methods have also been used. In the work of [82], a d-vector system is proposed. The d-vector is the averaged activations from the last hidden layer of a DNN that is discriminatively trained to predict speaker labels. The d-vector alone does not outperform i-vector system, but they are complementary. An end-to-end system for TD ASV was proposed in [72], where a long short-term memory (LSTM) neural network was applied because it has the ability to model the temporal structure of short utterances. This system outperformed the i-vector system and it should be interesting to compare it with other systems like GMM-UBM system and on other open databases. DNN based features were integrated into a GMM-UBM framework in [36]. The GMM-UBM system showed good results, which have proven hard to beat by most TD speaker verification research [12, 36, 81, 83]. All of these approaches model both the speaker identity and the lexical content of the passphrase. Different features were also investigated for TD systems, in [84], amplitude and phase-based features are combined together for short duration conditions, and showed that the fused system has better results, especially in female case.

Although text-independent systems are more challenging to implement, their potential for use in various applications makes them worthy of research. For TI short duration speaker verification, i-vector/GPLDA systems still serve as the state-of-the-art. In the development of speaker verification techniques, the long duration of enrolment and test utterances are preferable (hard to meet) condition. Unlike other biometrics like fingerprint, information from speech is accumulated with time, because speech is a time sequence, meaning that longer utterances have more information. Long duration utterances can be reasonably expected to cover all acoustic events such as vowels, and as such each one has more speaker discriminative information. An underlying assumption is that the contents of utterances (what has been spoken in utterance) are normalised by later transformations and representations. However, this assumption is not met by

short duration utterances. As the duration decreases, an utterance may not cover all the acoustic events. Consequently, some, but not all acoustic events are presented in the short duration utterances [85]. Additionally, if durations of utterances are long, each acoustic event is visited frequently, thus the relative amount of information in each acoustic event is not as diverse as in short duration utterance. How those acoustic events are visited in short duration utterances have strong influence on utterance representation and potentially makes it sensitive with contents.

To mitigate the problems introduced by short duration utterances, methods in the literature can be categorised into two branches. The first is devoted to compensating the mismatch caused by duration variation in the i-vector space. For example, score domain compensation for duration mismatch using a quality measure function (QMF), which takes durations of enrolment and test utterances into account, was introduced in [85]. The mismatch between long training and short test durations was compensated for, in the training phase for the total variability matrix and hyper-parameters of PLDA, by adding short utterances [86]. These techniques are proposed given the fact that i-vectors from long and short utterances do not have the same distribution.

The second branch of techniques aims to produce better representations of short duration utterances. Mismatch between long and short utterances arises primarily from the varying amounts of information in those utterances. However, the total variability model in the i-vector framework is trained on long utterance, which also contributes to the mismatch in i-vector space and provides less accurate representations of the utterances. Though the covariance of the i-vector posterior probability was integrated into the PLDA model in [87, 88], the computational load is larger, and modelling posterior covariance in a subspace may not be effective. A content aware local vector has been proposed [89]. Though different senones have been clustered agglomeratively in this method, it does not take into account the fact that different clusters may overlap. In [90], informative prior knowledge is used to compensate for channel variability, but

the prior assumption is still a Gaussian distribution. In [91], it is proposed to compensate phonetic information in the development of the total variability model, but it seems that using another latent variable to represent phonetic information is not optimal to remove the phonetic variation.

Several other works are related to phoneme matching. In [92], it was suggested that because i-vectors also contains other information such as transmission channel, acoustic environment or phonetic content, that i-vectors extracted from short utterances will be biased toward certain phonetic classes. Their experiments on both phoneme and duration mismatched, phoneme matched but duration mismatched, and phoneme and duration matched sets suggested that the duration and phonemes of the training and test data will affect the i-vector. By using the phonetic classification of the development data, the performance was improved. Other papers such as [93], defined a broad phoneme category to align the phoneme context of the training and test data, which was found to have some improvement.

In short duration ASV, the direct application of the DNN is much more challenging. As mentioned in Section 2.4.4, In [67], a DNN trained for an ASR system was used to replace the UBM in the ASV system. The UBM in ASV serves to generate a posterior probability for each frame. Although every Gaussian component may cover a specific acoustic region, they have no clear phoneme meaning, which makes it hard to do content matching in the test phase. Instead, in [67], a DNN is used to generate the posterior probability for each frame of the features, and then obtain zero- and first-order statistics. i-vector can be formed after this phase and followed by PLDA, similar to the standard i-vector/GPLDA system. In [94], a DNN with the same structure as in [67] was used, but content matching at the statistics level was proposed. In this case every node of the DNN has a specific phoneme meaning, e.g., a triphone. By aligning the zero- order statistics and scaling the first order statistics, some phonemes are scaled up means they are reused. If zero-order and first order statistics are diminished, then it means that

information of the correspondent phonemes is decreased in the training data. This method showed significant improvement in the condition where the content of test utterances is pronounced in the training data. However, in the more generative condition where the test prompts are not pronounced in the training data, the performance becomes worse. Besides replacing the UBM, the DNN can be used to generate features [95], such as bottleneck features and pcaDCT features [95]. Although the short duration case has been investigated for several years, the performance of short duration test at durations of 5 seconds and 3 seconds are still relatively poor [18, 94].

2.5 Database

Evaluating the system performance on a well-known database is necessary, so that researchers can compare their systems and methods. In this way, new methods with good results will be easily recognised by the community, which is important to help researchers come up with new ideas. The National Institute of Standards and Technology (NIST) is an organisation that makes contributions to many areas. The NIST speaker recognition evaluation (SRE) challenge is the main challenge workshop in the area speaker recognition, such as NIST SRE 2004, 2005, 2006, 2008, 2010 and 2012. For each of these challenges and workshops, NIST has released a database and evaluation plan [16]. These databases are the standard databases in the speaker recognition area. They contain many conditions including short utterance like 10 seconds. The NIST SRE databases are the main source of data for the research presented in this thesis, which will be used in the rest of thesis. Switchboard databases [96] including Switchboard II Parts 1, 2, 3, and Switchboard Cellular Parts 1 and 2 are also applied. For even shorter durations, e.g less than 3 seconds data, utterances can be generated by truncating the longer utterances to short ones. The PRISM database [97] will also be used to build and evaluate TI ASV systems. It contains clear, noise and reverberation conditions that are beneficial for this research. RSR2015 database [17] is used to develop models for TD speaker verification. The RedDot database [98]

is used for TD speaker verification and is used in Section 3.4. The RedDot consists of four parts: Part 1 is 'Common Pass-Phrase', in which every speaker has the same ten pass-phrases; Part 2, is 'Unique Pass-Phrase', where every speaker has ten different pass-phrases and there is no common pass-phrase between speakers; in Part 3, each speaker has two free-choice from several sentences; and Part 4 contains free text sentences that are unique across all sessions.

2.6 Performance Evaluation

In this research, the measurements of performance evaluation are equal error rate (EER) and minimum detection cost function (MinDCF). Figure 2.10 shows a detection error trade-off (DET) graph, which is a graphical plot of error rates for binary classification systems, plotting the miss rate vs. false acceptance rate. Figure 2.11 shows the confusion matrix of ASV. There are two types of errors in the performance evaluation, the miss rate (MR) and false alarm rate (FAR). These two error rates are functions of the decision threshold. The EER is determined to be the error rate at which the MR equals the FAR. EER is not an application measurement, while the MinDCF is an application dependent measurement defined as a weighted sum of the miss and false acceptance probabilities. The detection cost function is defined as

$$C_{Det}(\theta_t) = C_{Miss} \times P_{target} \times P_{Miss}(\theta_t) + C_{FAR} \times \left(1 - P_{target}\right) \times P_{FAR}(\theta_t) \quad (2\text{-}12)$$

where C_{Miss} and C_{FAR} are parameters of this cost function that will be provided by NIST SRE, P_{target} is the prior probability of the specified target speaker that will defined by organiser, θ is the threshold. As EER and MinDCF are generated by choosing particular points on DEV curve, it will be redundant to report all of them. Thus, if DET curve is provided to have a better visualisation of the performance, only EER will be reported as the numeric measurement.

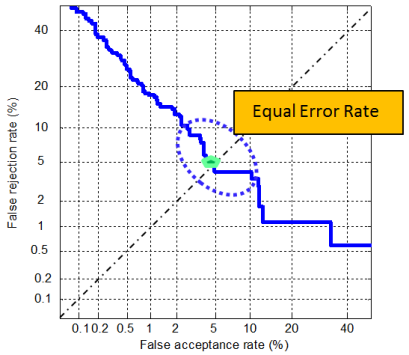


Figure 2.10 Detection error trade-off and Equal Error Rate.

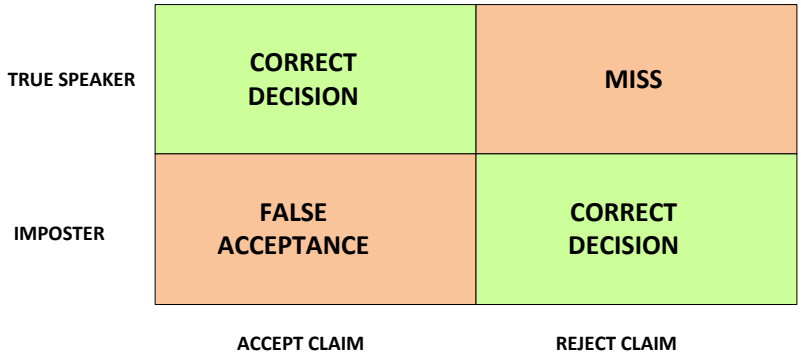


Figure 2.11 Confusion matrix of ASV.

2.7 Summary

In this chapter, an overall structure of speaker verification system is first introduced. In front-end, short-term spectral features like MFCCs are commonly used in recognition tasks in speech process. Different levels of features (i.e. short-term spectral features, prosodic features and high-level features) are also reviewed. Before feature modelling, feature normalisation techniques are applied to relieve channel mismatch and other mismatch problems in the feature domain. Different back-ends of automatic speaker verification systems have also been reviewed. Short duration speaker verification is then highlighted as a challenging problem that needs to be investigated. Finally, databases and evaluation metrics are presented. In the next few chapters, TD speaker verification is first discussed and followed by TI speaker verification.

3 PARALLEL SPEAKER AND CONTENT MODELLING FOR TEXT-DEPENDENT SPEAKER VERIFICATION

As mentioned in Section 2.4, a challenge faced by text-dependent speaker verification systems is that framing the alternative hypothesis is not straightforward. There are three potential alternative hypotheses, namely, that the speaker is not the claimed speaker but the pass-phrase is right ($H_{(\bar{\chi},\phi)}$), that the speaker is the claimed speaker but the test utterance is not the expected pass-phrase ($H_{(\chi,\bar{\phi})}$), and the speaker is not the claimed speaker and the test utterance is not the expected pass phrase ($H_{(\bar{\chi},\bar{\phi})}$) [79]. These three alternative hypotheses may be referred to as imposter-correct, target-wrong and imposter-wrong respectively. In this chapter, a parallel speaker and content modelling system is proposed to model the alternative hypothesis.

3.1 Modelling the alternative hypothesis

As mentioned above, a challenge faced by text-dependent speaker verification systems is how to effectively model the alternative hypothesis as there are two tasks in TD ASV. The primary task of interest is the verification of the speaker's identity and often a secondary task of interest is the verification of the lexical content of the passphrase. The combination of these two tasks utilised both person's knowledge of a piece of information ("something that you know"), and biometrics to verify identity of speaker. However, most systems in TD ASV still rely on speaker models (i.e, i-vector or GMM) that are adapted from background model (i.e. UBM) by spoken pass-phrase to capture passphrase information [17, 36, 81]. But the background model is trained to ignore passphrase information. Thus, it may not be optimal to capture information specified by the content of passphrase.

Unlike conventional systems, in this chapter, it proposes splitting the tasks of verifying the speaker identity and the lexical content running two systems in parallel to handle these two tasks in parallel before combining the results. Furthermore, we introduced a mixture selection method based on KL divergence to select discriminative mixtures in GMM for use in each speaker-passphrase model.

3.2 Proposed parallel system

The proposed system comprises of two sub-systems running in parallel, one that models speaker characteristics and verifies the speaker identity operating on the assumption that the right pass-phrase was spoken and a second one that models lexical content detects if the right pass-phrase was spoken as shown in Figure 3.1. Both sub-systems make use of the same front-end and the outputs of both sub-systems are combined to test against all three alternate hypotheses.

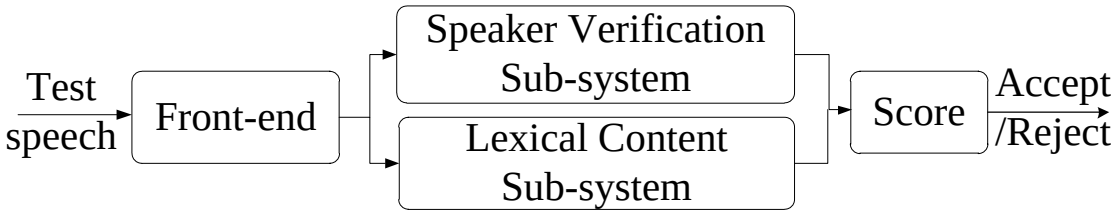


Figure 3.1 Proposed parallel speaker and content modelling.

The front-end of this system is comprised of standard MFCC features of 19 dimensions and the log-energy, along with their first and second derivatives. A vector quantisation model based voice activity detector was used [99] and Feature warping [45] was applied these features.

3.2.1 Proposed speaker verification sub-system

The speaker verification sub-system (denoted λ_{nHMM}, where n is the number of states in Hidden Markov model (HMM)) operates on the assumption that the lexical content is known for each trial in order to verify the claimed speaker identity. It employs HMM based speaker

models, where each state is represented by a suitable GMM, as shown in Figure 3.2. The n-state HMM is initialised with a UBM-GMM (λ_{UBM}) in each state and retrained with all data corresponding to each pass-phrase to estimate the background pass-phrase HMMs (λ_{BHMM}). The number of HMM is the same with the number of pass-phrase. Speaker specific passphrase HMMs (λ_{SPHMM}) are obtained via MAP adaptation of these background pass-phrase HMMs using examples of the target pass-phrase spoken by the target speaker.

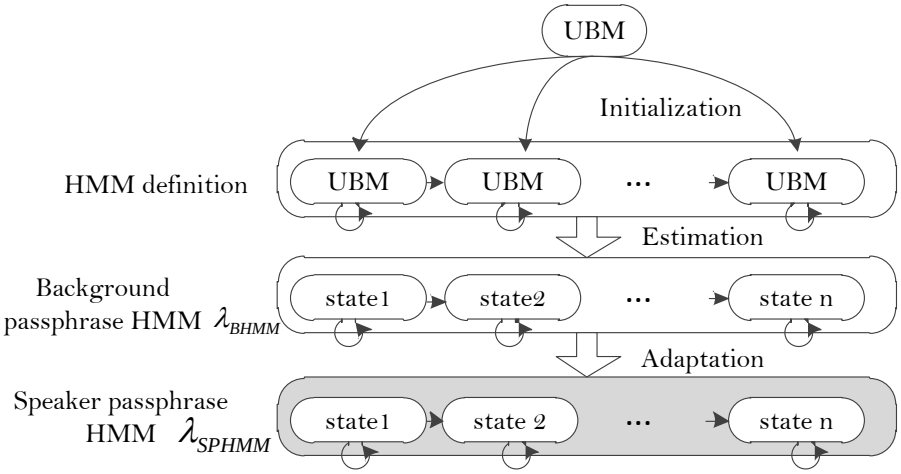


Figure 3.2 Proposed speaker verification sub-system using HMMs.

For score calculation, the averaged log-likelihoods $\log P(x|\lambda_{SPHMM})$ and $\log P(x|\lambda_{BPHMM})$ are calculated for each frame in test utterance, x, from λ_{SPHMM} and λ_{BPHMM} respectively using the Viterbi algorithm [100]. The final score for this sub-system is formulated as:

$$S_{HMM} = \log P(x|\lambda_{\mathrm{SPHMM}}) - \log P(x|\lambda_{\mathrm{BPHMM}}). \quad (3\text{-}1)$$

3.2.2 Proposed lexical content sub-system

The aim of the lexical content sub-system denoted as $\lambda_{n\mathrm{seg}}$, where n is the number of segments in the segment modelling, is to verify if the lexical content of test utterance matches that of the expected pass-phrase. One way to do that is to use HMM based methods as described in Section 2.1. However, as the number of sessions for each passphrase of one speaker is rather

limited, using HMM based methods to estimate the state-based alignment would not guarantee high accuracy. As a compromise, in the proposed system, an alternative approach utilising a left-to-right segment model is adopted.

The left-to-right segment model operates by splitting each pass-phrase into *S* segments and using a separate GMM to model each segment. Each segment GMM is expected to model the phonetic structure of the short segments of speech and the sequence of segment GMMs as a whole can be expected to model the overall temporal structure of the pass-phrase. i.e. two utterances that have the same phonetic content but with different phonetic order will not generate similar scores because the order of phonemes is different.

Figure 3.3 shows how the left-to-right segment model is created from a suitable universal background Gaussian mixture model (λ_{UBM}). Each utterance sequence from the same pass-phrase of a particular speaker is split into *S* segments of equal lengths. Feature vectors from each segment are used to adapt the background model λ_{UBM} to model that segment's lexical content and speaker information. The set of *S* adapted GMMs form the segment model of each pass-phrase for each speaker.

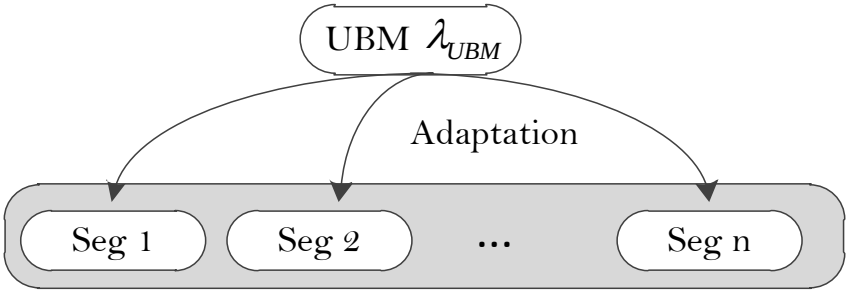


Figure 3.3 Proposed lexical content sub-system using segment models.

In the scoring phase, the test speech is divided into *S* segments and each segment is scored against the corresponding segment model and UBM to compute log-likelihoods from each

segment $\log P(x_i|\lambda_{Seg(i)})$ and for each frame in test utterance $\log P(x_i|\lambda_{UBM})$. The final score is then the mean of the log-likelihood ratio of each segment model:

$$Score_{seg} = \frac{1}{S}\sum_{i=1}^{S}\left(\log P\left(x_i|\lambda_{Seg(i)}\right) - \log P(x_i|\lambda_{UBM})\right) \quad (3\text{-}2)$$

where S is the number of segments, x_i denotes the i^{th} segment of speech and $\lambda_{Seg(i)}$ denotes the i^{th} segment GMM.

3.2.3 Score interpretation and combination

As previously mentioned, in text dependent speaker verification, the alternative hypothesis consists of three sub-hypotheses. In the proposed system, the speaker verification sub-system estimates the log-likelihood ratio of a model of the correct pass-phrase from the target speaker (λ_{SPHMM}) to a model of the correct pass-phrase from non-target speaker (λ_{BPHMM}). Consequently, the sub-system score S_{HMM} can be interpreted as comparing the hypothesis $\mathrm{H}_{(\chi,\not{p})}$ and $\mathrm{H}_{(\bar{\chi},\not{p})}$, i.e.

$$Score_{HMM} = log\, P\left(x|H_{(\chi,\not{p})}\right) - log\, P\left(x|H_{(\bar{\chi},\not{p})}\right). \quad (3\text{-}3)$$

The left-to-right segment models used in the lexical content sub-system differ for different pass-phrases and thus, pass-phrases that do not share the same lexical content will lead to low likelihood values even if they are from the same speaker because the temporal structure is different. Therefore, we can assume that λ_{seg} models both $H_{(\chi,\not{p})}$ and $H_{(\chi,\bar{\not{p}})}$ and denote it as $H_{(\lambda_{seg})}$. Finally, the λ_{UBM} is assumed to be text and speaker independent and the likelihood from the background model can be thought of as being represented by $P\left(x|H_{(\bar{\chi},\bar{\not{p}})}\right)$.

Adding the scores from the two sub-systems and using the interpretation above, the following equation can be obtained:

$$Score = \log P(x|H_{(\chi,\not p)}) - \log P\left(x\middle|H_{(\bar{\chi},\not p)}\right) + \log P(x|H_{(\lambda_{seg})} - \log P\left(x\middle|H_{(\bar{\chi},\bar{\not p})}\right). \quad (3\text{-}4)$$

Noting that $H_{(\lambda_{seg})}$ models $H_{(\chi,\not p)}$ and $H_{(\chi,\bar{\not p})}$, it is clear that the combined score covers all of the three sub-hypotheses.

3.3 Mixture selection

In speaker verification systems, including the proposed system, the UBM is assumed to be a text and speaker independent model that covers all imposters and lexical content. Since each pass-phrase is extremely short and phoneme coverage is limited in short duration text-dependent speaker verification [12], it is reasonable to argue that the number of adapted mixtures in each model is quite limited, which makes the models quite redundant as unadapted mixtures do not contribute to the scores. Moreover, if some mixtures are adapted based only on a small number of feature frames, this can lead to errors which could be reduced by removing those mixtures. In this section, we propose the use of a Gaussian mixture selection method to select the most discriminative mixtures between the UBM and adapted speaker GMM model.

A symmetric version of Kullback–Leibler (KL) divergence [101], called the Jensen–Shannon divergence, is used as a similarity measure between two Gaussian mixture models $f(x)$ and $g(x)$ (e.g., of the UBM and adapted speaker model). The Jensen-Shannon divergence can be calculated as

$$D(f,g) = \frac{1}{2}[D(f||g) + D(g||f)] \quad (3\text{-}5)$$

where $D(f||g)$ and $D(g||f)$ are the KL divergence between probability density function f to g, and g to f respectively. If Gaussian mixtures are assumed to have diagonal covariance matrix, the KL divergence between two mixtures is defined as in [102].

$$D(f||g) = \frac{1}{2}\left[(w_f - w_g)\, log\left(\frac{w_f}{w_g}\right) + \sum_{i=1}^{n}\frac{1}{2}(w_f\sigma_{fi}^2 - w_g\sigma_{gi}^2)\left(\frac{1}{\sigma_{gi}^2} - \frac{1}{\sigma_{fi}^2}\right) + \sum_{i=1}^{n}\frac{1}{2}\left(u_{gi} - u_{fi}\right)^2\left(\frac{w_f}{\sigma_{gi}^2} + \frac{w_g}{\sigma_{fi}^2}\right) + (w_f - w_g)\left(\frac{1}{2}\sum_{i=1}^{n}(\log\left(\sigma_{gi}^2\right) - \log\left(\sigma_{fi}^2\right)\right). \quad (3\text{-}6)$$

where n is the feature dimension; w_f and w_g are the weights; σ_{fi} and σ_{gi} are diagonal elements of the covariance matrix; u_{fi} and u_{gi} are elements of means of two mixtures. Assuming both mixtures have M components, the $M \times M$ KL divergence matrix is computed using equation (3-6).

Suppose there are M mixture components in both UBM and Speaker GMM, KL divergences between each mixture in UBM and speaker GMM will be calculated, which results in a $M \times M$ matrix M_{KL}. The KL divergence can be used as a measure between two distributions. If the distance between two distributions is further, the corresponding value of KL divergence will be larger. Discriminative mixtures are chosen based on this idea. First, the minimum element in the matrix is selected and the column index, which indicates the mixture place in UBM, and the row index, which indicates the mixture place of this element in the GMM, are recorded in two vectors V_{UBM} and V_{GMM}. Next, all elements in this column and raw are removed from M_{KL} as these two mixtures have been selected. This process is repeated till all elements are selected and the two vectors record all the mixtures in ascending order. This process makes the mixtures rank higher are more discriminative. Since the higher a mixture appears in V_{UBM} or V_{GMM}, the more discriminative that mixture must be, mixtures in towards the bottom of V_{UBM} and V_{GMM} will only be selected if the required number of mixtures is high enough. Systems with these mixture selections will be denoted as λ_{mM}, where m is the required number of mixtures. As the mixture selection will be applied to the lexical content sub-system, it will be denoted as λ_{mM_nS}.

3.4 Experiments and results

The baseline system is a GMM-UBM system which is mentioned in Section 2.4.1. Standard MFCC features of 19 dimensions with log-energy and their first and second derivatives were used. A vector quantisation model based voice activity detector was used [99]. Feature warping [45] was applied. A gender-dependent UBM of 512 Gaussian mixtures was created using all the utterances from male speakers in the RSR2015 database [103]. As number of female speakers in RedDot database is quite small (only 17 compared with 72 male speakers), experiments on male part are only conducted. The MFCC feature extraction and UBM training were done using the HTK toolkit [104]. The UBM was then adapted to each pass-phrase of speakers using maximum a posterior (MAP) algorithm and the corresponding enrolment utterances. Only the means of of the GMMs are adapted, while weights and covariances are shared across all models. In the rest of this section, this baseline system is referred as λ_{GMM}.

Experiments were conducted on the RedDots database [105]. This database was collected for short duration text-dependent speaker verification. As part 2 and part 3 of RedDot database are not common pass-phrase conditions (which means there is no λ_{BHMM} needed) and part 4 is not designed for common TI system, they are not suitable to validate the proposed methods and consequently only Part 1 was considered in this work. Test protocols were provided along with the RedDots database [105]. Results are reported for three different kinds of non-target trials (imposter-correct, target-wrong, and imposter-wrong) in terms of EER that are mentioned in Section 2.6 and Section 2.8. The MinDCF is not reported here as parameters for cost function are not defined in the RedDot database.

3.4.1 Parallel speaker and content modelling systems

A number of experiments using the two sub-systems described in Section 3.4 were carried out and the results are summarised in Table 3.1. When using four-segment left-to-right segment

models, target-wrong is improved substantially (45.6% relative improvement). This supports the assumption that the sub-hypothesis $H_{(\chi,\bar{\mathcal{P}})}$ is modelled by segment modelling in Section 3.4.1. However, the results of imposter-correct are degraded slightly and imposter-wrong are almost the same. This is not unexpected since the segment models have no mechanism of modelling the sub-hypothesis $H_{(\bar{\chi},\mathcal{P})}$. A model that takes this sub-hypothesis into consideration should be proposed. 8-segment λ_{8seg} left-to-right segmental models are also used, but the results are worse than those obtained with the 4-segment λ_{4seg} models. This may be because that the extremely short duration utterances contain limited phonemes and as such, having large number of segments becomes less useful.

Table 3.1 also reports the results obtained with the HMM based speaker verification sub-system described in Section 3.1.1. Experiments with 4 (λ_{4HMM}) and 8 (λ_{8HMM}) states (while keeping the total number mixtures in the HMM a constant) were carried out. The results showed that by using more states, the performance is slightly improved. Compared with the baseline λ_{GMM}, the results for imposter-correct is improved by 50%, while the results of target-wrong and imposter-wrong are degraded. This is due to the fact that the HMM based system was designed to model the sub-hypothesis $H_{(\bar{\chi},\mathcal{P})}$ only. When different passphrases are used in enrolment and testing, both the background HMM and speaker-dependent passphrase HMM are confused, and the other two sub-hypotheses are not taken into consideration by this sub-system which is mentioned before.

It can be seen from above two individual experiments that HMM based and segments models are complementary in terms of modelling the complete alternative hypothesis which contains three sub-hypotheses. Thus, it is natural to combine these two sub-systems. As the combined system models complementary alternative hypothesis, it can be expected to perform better than the baseline across all three metrics. The system column with the notation $\lambda_{8HMM} + \lambda_{4seg}$ lists the results of the combination of these two sub-systems. As analysed in Section 3.4.1, the scores

from different systems can be combined to cover the complete alternative hypothesis, as the summation of the log-likelihood of the three competing sub-hypotheses. It can be seen from the results that 26.7%, 46.2% and 22% relative improvement were obtained compared with the baseline, for imposter-correct, target-wrong and imposter-wrong respectively.

Table 3.1 Performance (EER%) of speaker verification sub-systems and lexical content sub-systems with different states and segments on Part 1 of RedDots database (male part only)

	Systems					
Alternate Hypotheses	λ_{GMM}	λ_{4seg}	λ_{8seg}	λ_{4HMM}	λ_{8HMM}	$\lambda_{8HMM} + \lambda_{4seg}$
imposter-correct	2.41	2.81	5.64	1.20	1.19	1.76
target-wrong	5.11	2.78	6.29	6.42	5.92	2.72
imposter-wrong	0.59	0.62	2.22	1.23	1.20	0.46

3.4.2 Incorporating mixture selection

Mixture selection was conducted by using the method introduced in Section 3.1.2. Table 3.2 shows the results of baseline and mixture selection systems. We can see that when only half of the mixtures (256) were chosen, the performances for the three different types of alternate hypotheses are improved. It was also observed that imposter-wrong trials will be better identified by the selected mixtures even when the number of mixtures decreases to 64, while the results for imposter-correct start to degrade below 128 mixtures. This observation suggests that information about lexical content can be represented by a limited number of discriminative mixtures (e.g., 64 compared with the original 512). This means that even though there are only a few frames aligned to a component, it may be discriminative in terms of speaker verification. This is likely to happen as there is limited speaker information in short duration utterances. When the number of mixtures falls to 32, performances are degraded for all three kinds of alternate hypotheses.

Given the results of the lexical content sub-systems in Table 3.2, it is clear that four segments are better than eight segments, so mixture selection is applied on this system to use half of number of mixtures in each model. These results are shown in Table 3.2 under the system column for λ_{256M_4S}. Compared with results without mixture selection, improvements across all three alternative hypotheses are obtained. Further combination with the speaker verification sub-system are 39.8%, 51.1% and 37.3% relative improvement for imposter-correct, target-wrong and imposter-wrong respectively. When compared with other approaches [106, 107], the results presented in this table also exhibit state-of-the-art performance. For example, in [107], the lowest EER for imposter-correct is 1.88%, while our methods lead to best score of 1.45%.

Table 3.2 Performance (EER%) of mixture selection with various mixtures on Part 1 of RedDots (male part)

	Systems						
Alternate Hypotheses	λ_{GMM}	λ_{256M}	λ_{128M}	λ_{64M}	λ_{32M}	λ_{256M_4S}	$\lambda_{256M\_4S} + \lambda_{8HMM}$
imposter-correct	2.41	2.34	2.50	2.96	4.34	2.80	1.45
target-wrong	5.11	4.50	3.98	4.18	5.62	2.50	2.50
imposter-wrong	0.59	0.48	0.52	0.77	1.24	0.56	0.37

3.5 Summary

In this chapter, we propose the use of two separate sub-systems, based on hidden Markov models and sets of segment GMMs, in parallel to model the combined speaker and lexical content information in short duration utterances. The performances of the individual sub-systems and that of the combined system are evaluated on the RedDots database and the two sub-systems are shown to be complementary. In addition, the use of a mixture selection method

is also proposed and the addition of the method to the overall system is shown to be beneficial. As stated in Section 2.7, next chapter will discuss the TI speaker verification.

4 MODEL COMPENSATION FOR SHORT DURATION SPEAKER VERIFICATION

In Chapter 3, we deal with the problem of TD ASV. As mentioned in Section 2.4, TI ASV is more challenging, and this thesis will focus primarily on TI ASV. In TI ASV, most state-of-the-art text-independent speaker verification systems consist of a total variability model, which models speaker and channel variability in a low-dimensional representation of speech utterances [19]. These are combined with Gaussian probabilistic linear discriminative analysis (GPLDA), which serves as a back-end to ASV system [20]. ASV systems conventionally require long enrolment utterances and operate on long test utterances (e.g., 2.5 minutes). In practical applications of ASV, short duration speaker verification would be significantly more desirable. However, as will be shown, the total variability model may not be optimal for short duration speaker verification. In this chapter, the i-vector representation extracted by the total variability model will be analysed and model compensation techniques proposed. The first method, described in Section 4.2, is a generalised form of total variability model, which intends to have better representation of utterance similar as i-vector. The second method in Section 4.3 is a local acoustic model, which is shown to be complementary to the total variability model.

4.1 Analysis of short duration utterance in the i-vector space

Section 2.4 highlighted that current start-of-the-art ASV system performance degrades sharply for short utterances. This section analyses how the duration of test utterances affects the i-vector representation.

To analyse the effects of short duration utterances, a preliminary experiment comparing zero-order statistics (herein referred to as N-vectors) as utterance representation to i-vectors with GPLDA based speaker verification system for long and short test utterances was conducted. N-vectors and i-vectors were estimated from utterances in the NIST SRE ‘04, ‘05, ‘06, ‘08, Switchboard II Parts 1, 2, 3, and Switchboard Cellular Parts 1 and 2. The NIST SRE’10 dataset consists of two conditions [13]: 8CONV-10SEC, in which the test utterances are of 10 seconds, and 8CONV-CORE, in which test utterances are around 2.5 minutes. Two additional conditions were created by truncating the 10 second test utterances to their first 5 and 3 seconds and are named as 8CONV-5SEC and 8CONV-3SEC, respectively. Their results are compared in Table 4.1. Only results for male condition are reported for simplicity. From these results, it can be seen that the performance of *N*-vectors is comparable with that of i-vectors for long duration utterances, but not in the case of the short duration utterances (10 seconds and less). Since the *N*-vectors (zero-order statistics) represent only the mixture occupancy of the UBM while the i-vectors are representative of the feature space distribution corresponding to the utterance using the UBM as prior information, *N*-vectors can be expected to be much more sensitive to variations in phonetic distributions across utterances. The results in Table 4.1 support the idea that zero-order statistics from long utterances are much more stable compared to those from short duration utterances, which lack phonetic diversity and have much less information in them. This is subsequently reflected in the i-vectors inferred from these statistics and consequently plays a significant role in the degradation of the performance of short duration text independent speaker verification systems.

A second observation about the effect of utterance duration can be made in terms of covariance matrices of the supervectors within the i-vector framework [108]. A basic tenet of the i-vector framework is that i-vectors are MAP estimates and the covariance, $\boldsymbol{B}$, of the supervector is related to the uncertainty of the estimated mean, M (Corollary 1 of [108]). The

larger the uncertainty, the less accurate the i-vector representing that supervector will be. Here, the trace of the covariance matrix is used as a measure of this uncertainty:

$$\bar{\sigma} = \frac{1}{R}\sum_{i=1}^{R} \mathrm{tr}(\boldsymbol{B_i}) \tag{4-1}$$

where $\mathrm{tr}(\cdot)$ is the trace operator, i is the number of utterances and R is the total number of utterances. The estimated uncertainty for utterances of 2.5 minutes, 10 seconds, 5 seconds and 3 seconds durations from the NIST SRE '04, '05, '06, '08 database described previously are given in Table 4.2. It can be observed that as the duration decreases, the uncertainty increases dramatically. Thus, it can be seen that the i-vector representation is not accurate for short duration utterances.

Table 4.1 Performance (EER %) using i-vectors and N-vectors on SRE'10 8CONV-CORE, 8CONV-10SEC, 8CONV-5SEC and 8CONV-3SEC conditions

Condition	EER (%)	
	i-vector	*N*-vector
8CONV-CORE	1.51	2.41
8CONV-10SEC	5.03	22.45
8CONV-5SEC	10.73	35.57
8CONV-3SEC	17.68	38.77

Table 4.2 Trace $\bar{\sigma}$ of the covariance matrix of supervectors in the i-vector framework

Measure	Duration			
	2.5MIN	10SEC	5SEC	3SEC
$\bar{\sigma}$	3.2	511.7	1107.2	1974.3

The mismatch of zero-order statistics between long and short duration utterances caused by insufficient phonetic content of short utterances is likely to be the reason for this uncertainty in the i-vector representation. The total variability matrix that maps the zero and first-order statistics of each component into the total variability space is trained on long utterances and fixed. When there is insufficient information pertaining to some components of the background model in short utterances, the distribution of all latent variables is influenced as a whole. For text-independent speaker verification, a technique that is able to detach and compare the component-wise information of utterances could be beneficial in this scenario and could complement the i-vector framework. This motivates the compensation techniques for models proposed in this chapter.

4.2 Proposed generalised variability model

From the literature, i-vectors [19] are widely used in speech processing tasks that involve pattern recognition, such as speaker verification [19], language identification [109], emotion recognition [110] and speech recognition [111]. Short-term spectral features of varying length utterances are efficiently represented as i-vectors through TVM. As mentioned in Section 2.4.3, TVM is a factor analysis model similar to probabilistic principal component analysis (PPCA) [59] in the aspect that latent variables are modelled as random variables drawn from a Gaussian distribution. The difference is that in TVM, supervectors are not found and are instead represented by Baum-Welch statistics estimated by a UBM. However, it may not be suitable to model the latent variables with a normal distribution in some scenarios, such as different genders [112]. In [108], it was pointed out that the assumption that latent variables were normally distributed which is equivalent to say the supervectors are also normally distributed may not be optimal. A distribution modelled as GMM, similar as in mixture of PPCA [113], should be beneficial if a large number of speakers are available for training purposes. However,

previous studies confine the development of its method to the case of the normal distribution [19, 108].

Because of the non-stationary nature of speech signals, distributions modelled by mixture of Gaussians are conventionally utilised to relax the assumption of a single normal distribution. For example, informative prior knowledge is used to deal with channel variability in the speaker verification area [90], but the prior assumption of latent variable is still a Gaussian distribution. Different mixture components are assigned to genders in [114] and multiple Gaussian components are used to tackle short duration [115, 116] and noisy conditions [117], although these techniques are used in i-vector space. The mixture of Gaussians used in prior is termed as MoG hereinafter. In language recognition, though a MoG was used in i-vector extraction stage in [118], parameters were estimated in the same way as a conventional TVM with a standard normal distribution, and a MoG prior was only used in the i-vector extraction stage. Thus, this section proposes the generalised variability model (GVM) to have MoG to model the distribution of latent variables.

4.2.1 Mixture of Gaussians distribution as prior

Let us begin with the assumption that the latent variables and supervectors do not follow a Gaussian distribution. Instead, MoG is used, with the assumption that each mixture is generated by a different source of variation. For example, to attack channel mismatch problem, modelling the distribution of latent variables as a mixture of Gaussians allows for each mixture component to represent a local cluster of latent variables corresponding to a different channel. Each source has its own prior and each prior bears the full responsibility of mapping a local cluster information (e.g., phoneme information or channel information) into different groups in the distribution of latent variable. The latent variable distribution is a combination of these different groups. The idea presented here is similar to independent factor analysis (IFA) [119]. However, comparing the two, each source is univariate in IFA, while here multivariate sources are

assumed. The posterior probability of each source given observed data is needed in the development of IFA and the proposed generalised variability model. It is calculated by the MoG itself in IFA. But in generalised variability model, the posterior probability of each source can be provided by a well-trained classifier (e.g., phoneme decoder). Figure 4.1 is an illustration of difference between TVM and the proposed GVM.

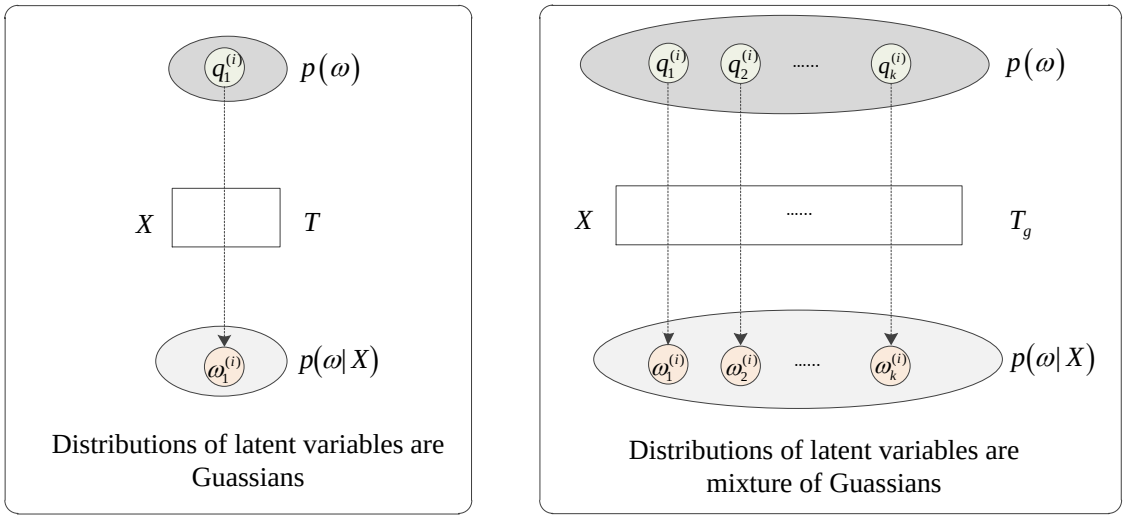


Figure 4.1 Comparison between TVM and GVM.

The generative equation of the proposed method is the same as (2-20) and the model is illustrated in the form of a graphical model in Figure 4.2. The latent variable is ω. Suppose K sources are specified in this model and we denote each source as q_k. The prior distribution of the latent variable ω is specified by a combination of K Gaussians and is given by

$$p(\omega) = \sum_{k}^{K} p(\omega|q_k)p(q_k) \tag{4-2}$$

where $p(q_k)$ is the prior distribution of source q_k. The probability of the latent variable ω given the source q_k is normally distributed with

$$p(\omega|q_k) = \mathcal{N}(m_k, \boldsymbol{B_k}) \tag{4-3}$$

and a flat prior of

$$p(q_k) = 1/K. \tag{4-4}$$

This means that there are several sources that generate the latent variables. For generating the latent variables, which source to choose depends on the probability that is provided by the prior MoG itself or a separate model. Note that if we constrain the prior distribution such that there is only one state and it has a standard normal prior, then it shrinks to the conventional TVM, meaning that conventional total variability model is a special case of this generalised one.

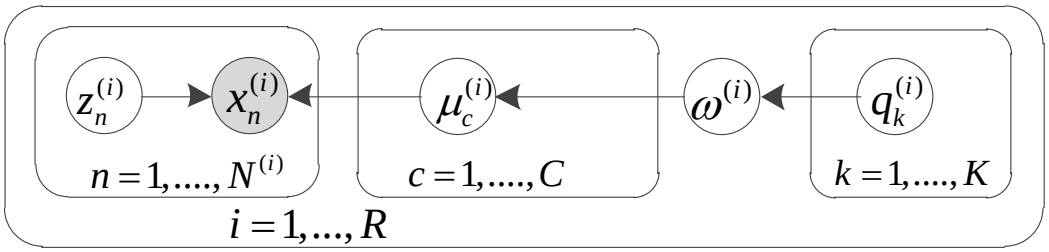


Figure 4.2 Graphical model of the proposed generalized variability model. The variables are: q - state variables. The indexes are: k - state index. The remaining symbols are the same as in Figure 2.12.

4.2.2 Posterior inferences

The collective feature frames for a given utterance is denoted as *X* (note that the superscript denoting utterance *i* is omitted in this subsection for brevity). In this model, the relationship of the latent variable *ω* to observable data *X* is the same as equation (2-20). The log-likelihood of the observed feature frames of one utterance is given latent variable and source is calculated as follows [108]:

$$\log[p(X|\omega,q_k)] = \log[p(X|\omega)] = \sum_{c=1}^{C}\left(N_c log \frac{1}{(2\pi)^{\frac{F}{2}}|\boldsymbol{\Sigma_c}|^{\frac{1}{2}}} \right.$$
$$\left. -\frac{1}{2}tr\left(\boldsymbol{\Sigma_c^{-1}S_c}\right) + \omega^* \boldsymbol{T_c^*\Sigma_c^{-1}}F_c - \frac{1}{2}N_c\omega^*\boldsymbol{T_c^*\Sigma_c^{-1}T_c}\omega \right) \tag{4-5}$$

where $\boldsymbol{S_c} = \sum_{n=1}^{N}(x_n - \mu_c)(x_n - \mu_c)^*$ and it is a matrix. The likelihood term is

$$p(X) = \sum\nolimits_k p(X|q_k)p(q_k) \tag{4-6}$$

where,

$$p(X|q_k) = \int p(X|\omega,q_k)p(\omega|q_k)d\omega = \int p(X|\omega)p(\omega|q_k)d\omega. \tag{4-7}$$

According to the graphical model in Figure 4.1, $p(X|\omega, q_k) = p(X|\omega)$, i.e. once ω is fixed, the identity of the state q is no longer relevant [119].

Directly optimizing the likelihood term $p(X)$ is hard to do as latent variables are inside the logarithm. Thus, the expectation-maximization (EM) algorithm [50] is used. The auxiliary function of the EM algorithm is similar with (2-13), and it is

$$Q(\theta,\theta_{old}) = \int p(H|X,\theta_{old})log[p(X|H)p(H)]dH \tag{4-8}$$

where H denotes the all latent variables including ω, z and q; X denotes the observed features, and θ_{old} denotes the model hyper-parameters from the previous iteration of the EM algorithm. The value of Q is used as a lower bound of the log-likelihood of observable data. It will increase with each iteration, leading to a local optimum of parameters [50].

As the mixture alignment of observed data and state posterior probabilities is not expected to change during parameter training, the posterior probability of latent variable to be estimated is

$$p(\omega|X) = \sum_k p(\omega|q_k, X)p(q_k) \tag{4-9}$$

where

$$p(\omega|q_k, X) \propto p(X|\omega, q_k)p(\omega|q_k)p(q_k). \tag{4-10}$$

by using the Bayes' theorem. The three terms on the right-hand side are given by equations (4-3), (4-4) and (4-5).

The posterior probability given the state and observed data is still a Gaussian. Substituting the right-hand side of equation (4-10) with equations (4-3) to (4-5), the first and second moments of the latent variables given the state and observed data are calculated as follows:

$$cov(\omega|q_k, X) = \left(\boldsymbol{B_k^{-1}} + \boldsymbol{T^*\Sigma^{-1}NT}\right)^{-1}, \tag{4-11}$$

$$E[\omega|q_k, X] = cov(\omega|q_k, X)\left(\boldsymbol{T^*\Sigma^{-1}}F + \boldsymbol{B_k^{-1}}m_k\right), \tag{4-12}$$

$$E[\omega\omega^*|q_k, X] = cov(\omega|q_k, X) + E[\omega|q_k, X]E[\omega^*|q_k, X] \tag{4-13}$$

where N, F and Σ are the stacked form of N_c, F_c and Σ_c, that the same are used as in [108], and $cov(\cdot)$ is the covariance operator.

According to [119], the following equation holds

$$E[f(\omega)|X] = \sum\nolimits_k p(q_k|X)E[f(\omega)|q_k, X] \tag{4-14}$$

where $p(q|X)$ is the posterior probabilities of state given the observed data. Taking $f(\omega) = \omega$ or $\omega\omega^*$, it is straightforward to calculate the expectations of latent variables given the observed data. Now, taking the first moment

$$E[\omega|X] = \sum\nolimits_k p(q_k|X)E[\omega|q_k, X]. \tag{4-15}$$

as the representation of a given utterance, we obtain the generalised i-vector, called the i_g-vector.

4.2.3 Parameter estimation

The labelling variable z can be omitted as it will not change across the training and inference stages. Starting from (4-8), the EM auxiliary function can be written as

$$\begin{aligned} Q(\theta, \theta_{old}) &= \sum_k \int p(\omega, q_k|X)\log[p(X|\omega, q_k)p(\omega, q_k)]d\omega \\ &= \sum_k p(q_k|X) \int p(\omega|q_k, X)\log[p(X|\omega)p(\omega|q_k)p(q_k)]d\omega. \end{aligned} \tag{4-16}$$

As the T matrix is the only parameter that needs to be estimated and it is included in $p(\omega|q, X)$ and $p(X|\omega)$, optimising the above equation is equivalent to optimizing

$$\begin{aligned} \hat{Q}(\theta, \theta_{old}) &= \sum_k p(q_k|X) \int p(\omega|q_k, X)\log[p(X|\omega)]d\omega \\ &= \sum_k p(q_k|X) E\big[\log[p(X|\omega)]\big]_{p(\omega|q_k, X, \theta_{old})} \end{aligned} \tag{4-17}$$

where $E(\cdot)_{p(\cdot)}$ is the expectation over distribution denoted $p(\cdot)$. By adding the utterance index back in, we arrive at

$$\hat{Q}(\theta, \theta_{old}) = \sum_i \sum_k p(q_k|X_i) E\big[\log[p(X_i|\omega_k)]\big]_{p(\omega_i|q_k, X_i, \theta_{old})}. \tag{4-18}$$

Substituting $log[p(X_i|\omega_k)]$ with (4-5), the auxiliary function then becomes

$$\begin{aligned} \tilde{Q}(\theta, \theta_{old}) = \sum_i \sum_k p(q_k|X_i) tr \Bigg(\Sigma^{-1} \Big(F_i E[\omega_i^*|q_k, X_i] \boldsymbol{T}^* \\ - \frac{1}{2} N_i \boldsymbol{T} E[\omega_i \omega_i^*|q_k, X_i] \boldsymbol{T}^* \Big) \Bigg). \end{aligned} \tag{4-19}$$

Setting the gradient of the expression regarding parameter T to 0, the following update equation can be obtained

$$\boldsymbol{T} = AC^{-1} \tag{4-20}$$

where

$$\boldsymbol{A} = \sum_i \sum_k p(q_k|X_i)E[\omega_i|q_k, X_i]F_i \tag{4-21}$$

$$\boldsymbol{C} = \sum_i \sum_k p(q_k|X_i)\boldsymbol{N_i}E[\omega_i\omega_i^*|q_k, X_i]. \tag{4-22}$$

The MoG prior can be updated from the original (4-3) simultaneously as follows

$$m_k^* = \frac{1}{R}\sum_i E[\omega_i|q_k, X_i], \tag{4-23}$$

$$\boldsymbol{B_k^*} = \sum_i cov(\omega_i|q_k, X_i)\,. \tag{4-24}$$

With those updating formulas, this algorithm can be trained iteratively.

4.3 Incorporating local acoustic information into the total variability model

In Section 4.2, a generalised variability model which is a generalised version of total variability model, is proposed. This model works for both long duration and short duration utterances. In this section, a local acoustic model which is complementary with above mentioned total variability model is proposed.

In recent years there has been increasing interest in short duration text-independent speaker verification systems, almost all of which focuses on the aforementioned i-vector/PLDA approach. For example, in [87, 88, 120], the covariance of the i-vector posterior probability was propagated to the PLDA model. In [85], score domain compensation for duration mismatch using a Quality Measure Function (QMF) that takes the durations of enrolment and test utterances into account was introduced. In [121], the mismatch between long enrolment and short test durations was compensated in the training phase of the total variability matrix and hyper-parameters of PLDA by adding short utterances.

As described in [108] and mentioned in Section 2.3.3, the idea behind i-vector modelling is that supervector representations of utterances can be mapped to a low dimensional space with little loss of accuracy. One of the main advantages of the i-vector framework is that channel variability can be compensated using techniques such as LDA, WCCN and PLDA in this low-dimensional space. However, as the duration of utterances decreases, the uncertainty of the i-vector representation increases. Speaker verification performance degrades sharply once the test utterance durations falls below 10 seconds [18], as was discussed in Section 4.1. It was shown that the basic GMM-UBM based methods are superior to subspace methods [36, 103, 122] when addressing text-dependent speaker verification with extremely short utterance (e.g., 3 seconds).

Supervectors can be regarded as representations of GMMs that differ only in their mixture means [123] and since the total variability model may describe inaccurate representation for short durations utterances, direct modelling of the supervectors may be beneficial. In [124, 125], parameter tying across mixtures in the total variability model is relaxed and banks of local variability vectors or concatenated local vectors are obtained. GPLDA was then trained on top them. The solution of modelling local acoustic variability proposed in this section is different to the above approach. In this section, as the uncertainty of the i-vector representation increases

sharply for short duration utterances, the latent variable model is bypassed, and local acoustic variability information is directly captured in the supervector space. The information in each phonetic group, referred to as local acoustic variability, is complementary to the total variability model. Following this, different weighting strategies are applied in order to take the relative reliability of local acoustic information into account.

4.3.1 Proposed local acoustic variability model

In short duration speaker verification, both channel variability and phonetic variability should be considered. Here it is proposed to apply GPLDA in the supervector space to model the channel variability with a semi-diagonal assumption to capture and compare local acoustic variability.

Given a collection of supervectors, $\mathcal{D} = \{\mathcal{M}_{ij}; i = 1,2,\cdots,S; j = 1,2\cdots,R_i\}$, where $\mathcal{M}_{\mathrm{ij}}$ is the supervector (after centring) corresponding to the j^{th} utterance from the i^{th} speaker, S is the number of speakers, R_i is the number of utterance of i^{th} speaker, the generative model can be described as:

$$\begin{bmatrix} \mathcal{M}_{i1}{}^{*} \\ \vdots \\ \mathcal{M}_{iR_i}{}^{*} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V} \\ \vdots \\ \boldsymbol{V} \end{bmatrix} z_i + \begin{bmatrix} \bar{\varepsilon}_{i1} \\ \vdots \\ \bar{\varepsilon}_{iR_i} \end{bmatrix} \tag{4-25}$$

where V is a factor loading matrix of $CD_f \times D$ dimension C is the number of mixture component in UBM, D_f is the dimension of feature frame, D is the dimension of z_i, z_i is a vector of latent variables which have a standard Gaussian distribution, $N(0, \boldsymbol{I})$, and $\bar{\varepsilon}$ is a residual term that is assumed to be Gaussian with zero mean and a covariance matrix denoted by $\bar{\Sigma}$ of which dimesion is $CD_f \times CD_f$. The first two moments of the latent variables are then calculated as follows:

$$E(z_i) = (\boldsymbol{I} + R_i \boldsymbol{V}^{*} \bar{\boldsymbol{\Sigma}}^{-1} \boldsymbol{V})^{-1} \sum_j \boldsymbol{V}^{*} \bar{\boldsymbol{\Sigma}}^{-1} \mathcal{M}_{ij}\,, \tag{4-26}$$

$$E(z_i z_i^*) = \left(\boldsymbol{I} + R_i \boldsymbol{V}^* \overline{\boldsymbol{\Sigma}}^{-1} \boldsymbol{V}\right)^{-1} + E(z_i) E(z_i^*). \tag{4-27}$$

Here it is assumed that the covariance is block diagonal, which, in addition to reducing the computational burden, imposes the underlying assumption that the mixture components are independent while preserving covariance information within the components (local covariance). Using the matrix inverse lemma [126], the estimation of the moments of the latent variables can then be broken down as follows

$$E(z_{ic}) = (\boldsymbol{I} + R_i \boldsymbol{V_c^*} \overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}^{-1} \boldsymbol{V_c})^{-1} \sum_j \boldsymbol{V_c^*} \overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}^{-1} \mathcal{M}_{ijc} \tag{4-28}$$

$$E(z_{ic} z_{ic}^*) = (\boldsymbol{I} + R_i \boldsymbol{V_c^*} \overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}^{-1} \boldsymbol{V_c})^{-1} + E(z_{ic}) E(z_{ic}^*) \tag{4-29}$$

where $\boldsymbol{V_c}$ is the $D_f \times D$ dimensional sub-matrix of $\boldsymbol{V}$ corresponding to the c^{th} Gaussian mixture component of the UBM, the subscript c denotes the c^{th} block of the corresponding parameter or variable.

Given an enrolment supervector $\mathcal{M}_e$ and a test supervector $\mathcal{M}_t$ from a trial, the score is calculated as follows:

$$Score(\mathcal{M}_e, \mathcal{M}_t) = \sum_c (log\, S_{1c} - log\, S_{0c}) \tag{4-30}$$

where $S_{1c.}$ and S_{0c} are scores for hypothesis that $\mathcal{M}_{ec}$ and $\mathcal{M}_{tc}$ are from the same speaker and different speakers respectively, and calculated as follows:

$$S_{1c} = \mathcal{N}\left(\begin{bmatrix}\mathcal{M}_{ec}\\ \mathcal{M}_{tc}\end{bmatrix}; 0, \begin{bmatrix}\boldsymbol{V_c V_c^*} + \overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}} & \boldsymbol{V_c V_c^*}\\ \boldsymbol{V_c V_c^*} & \boldsymbol{V_c V_c^*} + \overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}\end{bmatrix}\right), \tag{4-31}$$

$$S_{0c} = \mathcal{N}\left(\begin{bmatrix}\mathcal{M}_{ec}\\ \mathcal{M}_{tc}\end{bmatrix}; 0, \begin{bmatrix}\boldsymbol{V_c V_c^*} + \overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{V_c V_c^*} + \overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}\end{bmatrix}\right). \tag{4-32}$$

The parameter estimation and scoring can be processed in a component-wise manner. The EM algorithm is used to estimate the hyper-parameters. A comparison between the i-vector/GPLDA system with the proposed method is presented in Figure 4.3.

4.3.2 Likelihood weighting

In the framework proposed in Section 4.3.1, during the hyper-parameter training stage, the mean vector $\mathcal{M}_{ijc}$ of c^{th} component of the supervectors corresponding to each session is treated identically across the training data. This is not a satisfactory assumption as the number of frames aligned to each component of the UBM is not equal for each utterance. To remedy this, the EM algorithm's M-step described is modified to take into account the relative reliability of the mean vectors.

Let $\theta_c = \{V_c, \bar{\Sigma}_c\}$ denote c^{th} block of the parameters that need to be estimated. In the M step, the log likelihood is to be maximised. The auxiliary function is

$$Q(\theta_c'|\theta_c) = E_z\{logP(\mathcal{D}_c, Z_c|\theta')|X, \theta\} = \sum_i \sum_j \mathcal{L}_{ijc} \tag{4-33}$$

where $E_z(\cdot)$ denotes the expectation over distribution of z and

$$\mathcal{L}_{ijc} = E_z\left\{\left[log\frac{1}{(2\pi)^{\frac{D_f}{2}}|\bar{\Sigma}_c|^{\frac{1}{2}}} - \frac{1}{2}\left(\mathcal{M}_{ijc} - V_c z_{ic}\right)^T \bar{\Sigma}_c^{-1}\left(\mathcal{M}_{ijc} - V_c z_{ic}\right) - \frac{1}{2} z_{ic}^* z_{ic}\right]\right\}. \tag{4-34}$$

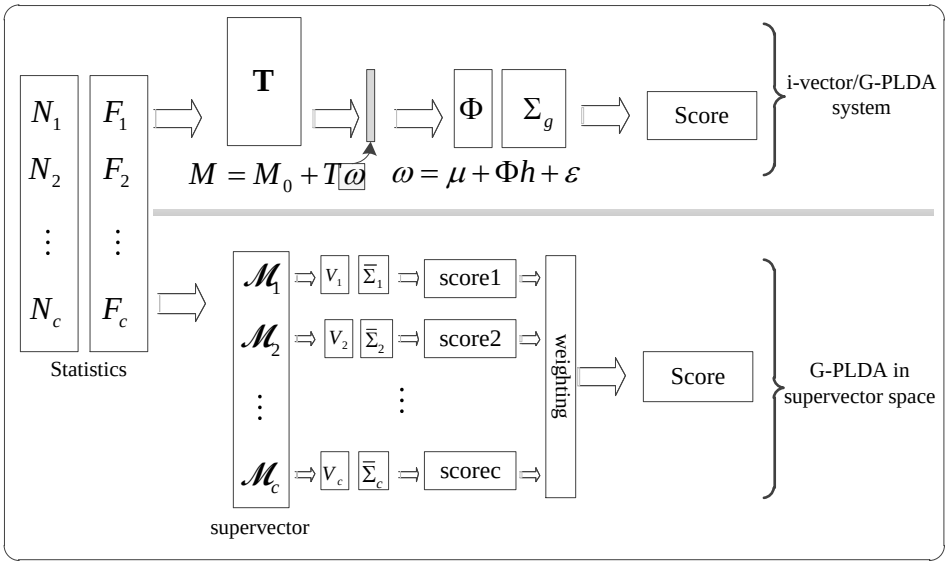


Figure 4.3 Comparison between i-vector/GPLDA system (upper panel) and GPLDA in supervector space system (lower panel).

For each speaker, we weight the likelihood of each utterance from the speaker as

$$\mathcal{L}_{i.c} = \frac{R_i}{\sum_j N_{ijc}} \sum_j N_{ijc} \mathcal{L}_{ijc} \tag{4-35}$$

where N_{ijc} is the zero-order statistics aligned to the c^{th} component of UBM from j^{th} session of i^{th} speaker.

The interpretation of this weighting is that the relative importance of i^{th} speaker to the total log likelihood is valued by the session number R_i, while the relative importance of each session is proportional to the factor $N_{ijc}/\sum_i(N_{ijc})$, which reflects the reliability of the mean vector of this session. The updated parameters will favour those sessions that have more frames aligned to c^{th} component as they should be more robust. Thus, the solution to maximise the auxiliary function is:

$$V_c = \left(\sum_{\boldsymbol{i}} \frac{R_i}{\sum_j (N_{ijc})} \sum_j N_{ijc} \mathcal{M}_{ijc} E(z_i)^* \right) \left(\sum_i R_i E(z_i z_i^*) \right)^{-1}, \tag{4-36}$$

$$\bar{\Sigma}_c = \frac{1}{\sum_{\boldsymbol{i}} (R_i)} \left(\sum_{\boldsymbol{i}} \frac{R_i}{\sum_j (N_{ijc})} \sum_j N_{ijc} \left(\mathcal{M}_{ijc} \mathcal{M}_{ijc}^* - \boldsymbol{V_c} E(z_i) \mathcal{M}_{ijc}{}^* \right) \right). \tag{4-37}$$

4.3.3 Mean vector weighting

In the E step of the EM algorithm, all sessions from one speaker are used to estimate the posterior probability of the latent variables. In (4-28), mean vectors from different sessions are summed to estimate the posterior mean. But mean vectors from different sessions are not equally reliable. Thus, a weighting similar to the proposed likelihood weighting from Section 4.3.2 can be applied to the posterior probability estimation. The idea is that mean vectors, $\mathcal{M}_{ijc}$, from the same speaker should not be regarded identically. Recall the calculation of zero-order statistics in Section 2.4.3, the larger the value of N_{ijc}, the more reliable the corresponding mean vector is, and should be assigned a higher weight when estimating the posterior probability. Specifically, each mean vector is weighted by its corresponding zero-order statistic in the E step. The revised mean of the posterior distribution is now

$$E(z_{ic}) = (\boldsymbol{I} + R_i \boldsymbol{V_c^*}^{\top} \overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}^{-1} \boldsymbol{V_c})^{-1} \frac{R_i}{\sum_j (N_{ijc})} \sum_j N_{ijc} \boldsymbol{V_c^*}^{\top} \overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}^{-1} \mathcal{M}_{ijc}. \quad (4\text{-}38)$$

The covariance of the posterior probability can also be modified to take into account the relative reliability of the mean vector, however since the covariance of posterior probability is excluded from the estimation of the expectation of the latent variable, it is not beneficial to perform weighting in covariance of posterior probability. Thus, the second moment of latent variable has the same form as (4-29).

4.3.4 Score weighting

As per equation (4-30), the GPLDA score in the supervector space with block diagonal covariance assumption is the summation of the sub-scores of each component of UBM. A question can again be raised about whether these sub-scores are equally important. It is reasonable to assume that the relative reliability of a sub-score should be taken into account by weighting them by the number of frames that are aligned with the corresponding component of the UBM. Since the enrolment data comprises of long utterances, the sub-scores are only weighted by the short test utterances. The proposed sub-score weights are as follows:

$$\gamma_c = \frac{N_{tc}}{\sum_c (N_{tc})} \quad (4\text{-}39)$$

where t denotes corresponding variable or parameter is from test utterance. Including these weightings into equation (4-30), the final score is then calculated as

$$Score(\mathcal{M}_e, \mathcal{M}_t) = \sum_c \gamma_c (log\, S_{1c} - log\, S_{0c}). \quad (4\text{-}40)$$

4.4 Experiments

4.4.1 Experiments and discussion of GVM

Experiments were conducted to validate the effectiveness of the proposed generalised variability model. Specifically, an i-vector/GPLDA system using the proposed model is compared with the

conventional i-vector/GPLDA systems that uses the standard TVM on NIST SRE 2010 [13] common condition 5 (CC5) which consists of CORE-CORE extended (denoted as CORE-EXT), CORE- CORE, and CORE-10SEC.

For the generalised variability model system (denoted as GVM for simplicity), standard MFCC features of 13 dimensions with their deltas and delta-deltas were used in conjunction with a vector quantisation model based a voice activity detector [99] followed by feature warping [45]. Gender-dependent UBMs of 1024 Gaussian mixtures with diagonal covariance were created using utterances from background data from the NIST SRE'04, 05, 06, 08, Switchboard II Part 1, 2, 3 and Switchboard Cellular Part 1 and 2 datasets. One utterance was chosen from each speaker's available data to retain speaker diversity while reducing the overall amount of data [127]. The generalised version of T matrix was estimated by (4-20) and priors were updated simultaneously with (4-23) and (4-24). LDA was then applied to reduce the dimension to 200. The i-vectors were then radially Gaussianised followed by length normalisation as described in [64].

For the conventional TVM (denoted as Gaussian), the same MFCCs, were used. The same sets of development, training and test data were employed for the baseline and proposed systems. The same UBM was used as well. T matrices of rank 400 were estimated by using the MSR Toolbox [128].

A PLDA model was adopted as the back-end for both systems on top of i-vectors and i_g-vectors. i-vectors and i_g-vectors from background datasets were estimated and used to train each GPLDA. The dimensionality of the speaker factors for both systems was set as 200.

In these experiments, phonetic groups are used as a source of variation in the experiments for illustration (other sources of variations can also be modelled if desired). The BUT phoneme decoder [129] is used to obtain phonetic posterior probabilities in this work. To further simplify

the system, similar phonemes (e.g., long and short duration phonemes) are clustered, resulting in 14 phonetic groups. The phonetic group information is presented in Table 4.3. Note that the characters used in this table are phonetic labels [129]. The corresponding phonetic posterior probabilities are then summed over the groups. In the generalised variability model, the priors were estimated in the following way. First, the same TVM as the baseline was used. Utterances in the background datasets were aligned with the phoneme decoder and UBM. Phonetic based zero- and first-order statistics are calculated as $N_{kc} = \sum_n p(k|x_n)p(c|x_n)$ and $F_{kc} = \sum_n p(k|x_n)p(c|x_n)x_n$. Phonetic vectors were estimated based on these phonetic statistics. One Gaussian was then assigned to each phonetic group to fit the vectors. The MoG prior is obtained as per (4-23) and (4-24).

Table 4.4 summarises the experimental results in terms of EER and MinDCF. The system with a standard Gaussian prior is denoted as TVM and system with the mixture of Gaussians prior is as GVM. From the table, it can be seen that the GVM system has better EER for all conditions except CORE-CORE condition, while improvements were observed for all conditions with a mixture of Gaussians as prior in terms of MinDCF. In general, relative improvement around 5% to 15% across gender and different conditions were observed by replacing standard Gaussians with MoG.

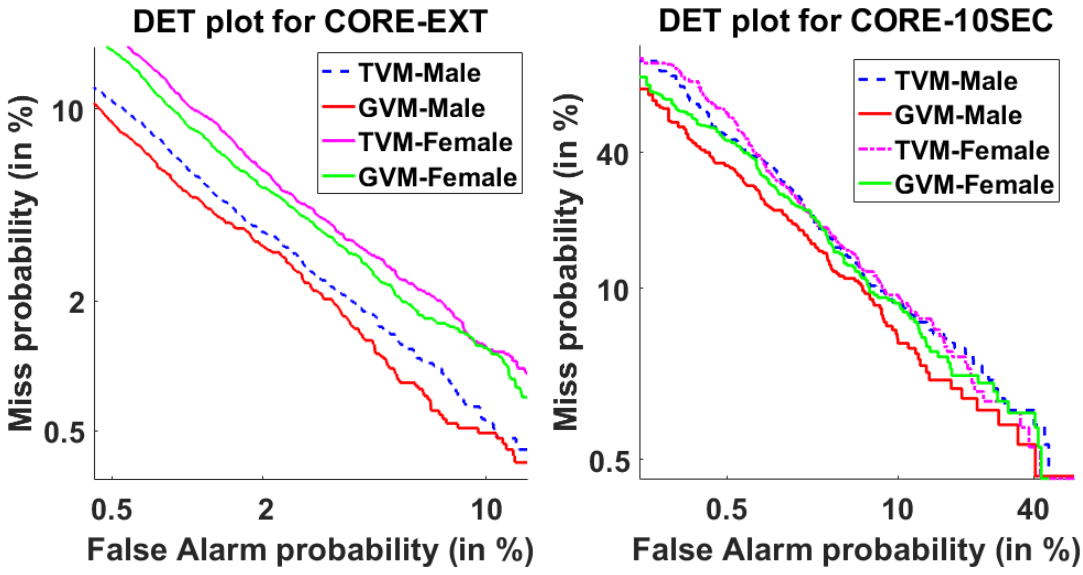
Figure 4.4 shows the DET curves of CORE-EXT and CORE-10SEC conditions. The DET curves confirm that using MoG as a prior reduces both false alarm and missing probabilities. From these results, it is also observed that shorter conditions like CORE-10SEC benefit more from the MoG prior. This may be due to the fact that mixture components in the prior have phonetic meaning and will remove some phonetic nuisance information, resulting in an i_g-vector that is less sensitive to phonetic variations that are more severe in shorter utterances.

Table 4.3 Phonetic group information

Phonetic group	Phonemes
Group 1	J, J:, m, m:, N, n, n:
Group 2	b, b:, d, d_, d_:, g, k, k:, l, l:, p, t, t:, x
Group 3	dz, tS, tS_, ts, ts_, t1, t1:
Group 4	F, f, h, h1, S, S:, s, s:, Z, z, z:
Group 5	r, r:
Group 6	j, j:
Group 7	A:
Group 8	E, e:
Group 9	i, i:
Group 10	O, o, o:
Group 11	:2, _2
Group 12	u, u:
Group 13	v
Group 14	y, y:

Table 4.4 Performances (EER% and MinDCF%) of standard TVM, and GVM systems on the NIST SRE'10 CORE-EXT, CORE-CORE, and CORE-10SEC sets with CC5 conditions

System	**EER (%)**					
	Male			Female		
	CORE -Ext	CORE - CORE	CORE -10SEC	CORE -EXT	CORE - CORE	CORE -10SEC
TVM	2.80	2.85	8.75	3.70	3.19	9.19
GVM	2.61	2.90	7.41	3.51	2.94	8.32
MinDCF(%)						
TVM	8.28	8.19	25.93	10.97	9.60	27.56
GVM	7.56	7.40	21.58	10.22	8.29	24.71



a *b*

Figure 4.4 DET plot of two conditions a) CORE-EXT, and b) CORE-10SEC.

4.4.2 Experiments of local acoustic model

As the local acoustic model is proposed for short duration speaker verification, the 8CONV-10SEC condition of the NIST SRE'10 [13] was chosen for these experiments along with the 8CONV-5SEC and 8CONV-3SEC conditions, which are designed for short duration ASV. The baseline system is an i-vector/G-PLDA system that is the same with Section 4.4.1. For the proposed method and baseline, identical MFCC features and UBM were used with Section 4.41, as well as identical development. In addition to the baseline i-vector/GPLDA system, the proposed system is also compared to a local variability model (LVM) system [124] utilising the same front-end and UBM as the proposed technique.

Table 4.5 summarises the performances of the i-vector/GPLDA baseline system, LVM system and proposed system. The term S-GPLDA is used to represent the proposed GPLDA in supervector space with block diagonal covariance assumptions without any of the weighting techniques presented in Sections 3.3.1 to 3.3.4. Compared to the baseline, there is gap between

the proposed system and the baseline. However, we also observe that as the duration of test utterance decrease, the gap tends to be smaller.

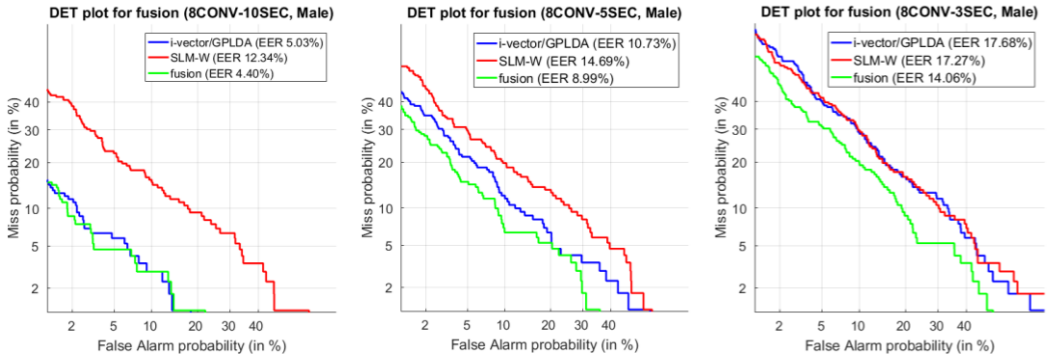
Table 4.5 Performance (EER%) of the baseline system, proposed and fusion systems on the NIST SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions

System number		Male			Female		
		10SEC	5SEC	3SEC	10SEC	5SEC	3SEC
1	Baseline	5.03	10.73	17.68	6.16	12.43	18.90
2	S-GPLDA	14.52	18.85	22.27	17.44	20.95	25.50
3	S_W	13.49	15.81	18.14	14.16	18.05	19.64
4	SL_W	12.33	15.46	18.47	11.96	16.19	19.39
5	SLM_W	12.34	14.69	17.27	12.18	16.00	18.76
6	LVM	9.60	16.57	22.76	11.34	17.98	22.95
1+5	Fusion1	**4.40**	**8.99**	**14.06**	**5.92**	**11.24**	**15.31**
1+6	Fusion2	4.65	10.20	16.28	6.07	11.53	17.85

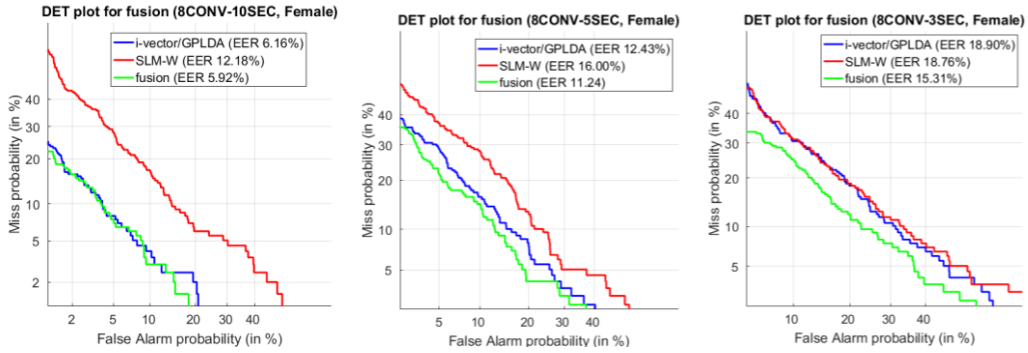
Systems denoted by S_W, SL_W and SLM_W are the proposed method with additional score weighting (Section 3.3.4), score and likelihood weighting (Section 3.3.2), and score, likelihood and mean vector weighting (Section 3.3.3) respectively. Based on the results in Table 3.5 it can be seen that when all three weighting techniques were used, for 10 seconds and 5 seconds test conditions, the performance of the proposed approach was still inferior to those of the baseline system. However, the gaps again decreased as the duration of test utterance decreased and for 3 second test condition a slight improvement was observed for both male and female speech over the baseline. Compared with the LVM system [124], the SLM_W system obtained superior performances on 5 seconds and 3 seconds conditions for both male and female conditions, but not in the longer 10 second test condition.

Given that the proposed GPLDA in supervector space was designed to capture local acoustic variability to complement the total variability framework of the baseline i-vector system, the baseline and the proposed system can be expected to be complementary and fuse well. In the experiments reported in this work, these systems are fused at the score level. Scores from the baseline system and the proposed system with all three proposed weighting techniques (SLM_W) are fused using the BOSARIS Toolkit [130] and denoted as Fusion1. Based on the results it is clear that the two approaches are complementary and the fusion leads to substantial improvements, particularly in the 3 second test condition. This is supported by DET plot in Figure 4.5. The baseline was also fused with LVM system (denoted as Fusion2) and compared to the proposed system. It can be seen that the proposed system outperformed LVM system when fused with baseline under all test conditions.

It can be seen that the local acoustic model is complementary to the total variability model. The experimental results also suggest that, the proposed local acoustic information model (LSM_W) complements the traditional total variability space modelling approach by incorporating local acoustic variability information with greater benefits being observed for shorter test utterances.



(a) *(b)* *(c)*



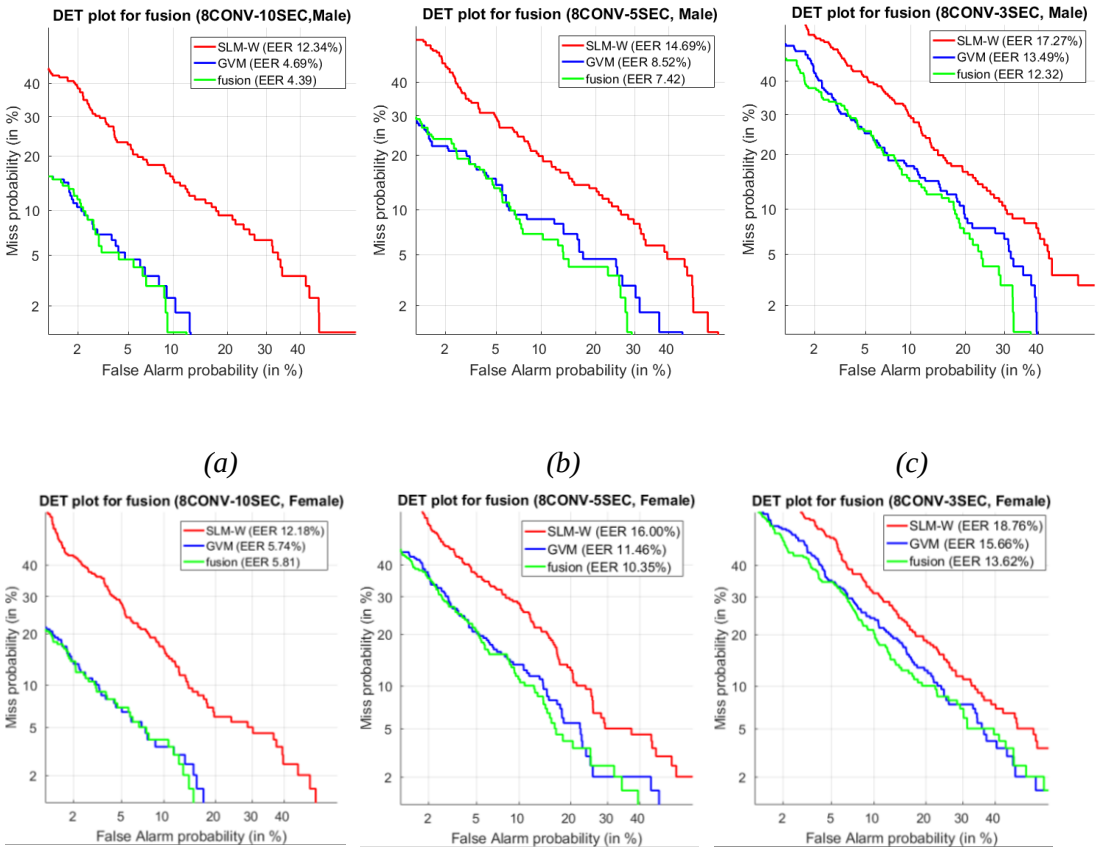
(d) *(e)* *(f)*

Figure 4.5 DET plot of systems (i-vector/GPLDA, SLM_W, and fusion) for (a) 8CONV-10SEC, Male, (b) 8CONV-5SEC, Male, (c) 8CONV-3SEC, Male, (d) 8CONV-10SEC, Female, (e) 8CONV-5SEC, Female, and (f) 8CONV-3SEC, Female.

The GVM system is also presented here as a better utterance vector representation than i-vector. Table 4.6 and Figure 4.6 summarise the results. It can be seen that better performance can be obtained with GVM system and lead to higher improvement when fused with the local acoustic model which is denoted as the Fusion 3 in Table 4.6. As can be seen in Figure 4.5, the gaps between fusion systems denoted by green line in each sub-figure and GVM system denoted by blue line are not as large as those in Figure 4.5. This suggests that GVM can generate better vector representation of utterance than TVM. This i_g-vector may cover more information and thus makes the addition of local acoustic information less useful.

Table 4.6 Performance (EER%) of the baseline system, proposed and fusion systems on the NIST SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions

System number		Male			Female		
		10SEC	5SEC	3SEC	10SEC	5SEC	3SEC
1	GVM	4.69	8.52	13.49	5.74	11.46	15.66
5	SLM_W	12.34	14.69	17.27	12.18	16.00	18.76
1+5	Fusion3	**4.39**	**7.42**	**12.32**	**5.81**	**10.35**	**13.62**



(a) *(b)* *(c)*

(d) *(e)* *(f)*

Figure 4.6 DET plot of systems (GVM, SLM_W, and fusion) for (a) 8CONV-10SEC, Male, (b) 8CONV-5SEC, Male, (c) 8CONV-3SEC, Male, (d) 8CONV-10SEC, Female, (e) 8CONV-5SEC, Female, (f) 8CONV-3SEC, Female.

4.5 Summary

In this chapter, the i-vector representation is analysed for short utterances and it is found that the i-vector representation is not accurate for these short utterances. On one hand, the conventional i-vector representation restricts the prior distribution to be a standard Gaussian which is not optimal in some situations. Arbitrary distributions can instead be achieved with a mixture of Gaussians, and so the conventional i-vector is generalised to the proposed i_g-vector. The conventional total variability model is shown to be a special case of the proposed generalised variability model. On the other hand, local acoustic variability is also proposed as a complementary system to complete the total variability model for short utterance conditions.

The local acoustic variability model is shown to be highly complementary to the total variability model. The proposed models are validated on NIST SRE 2010 databases. Experimental results show that better performances can be obtained in all conditions. In the next chapter, the phonetic properties of short duration utterances will be examined, and phonetic based methods are proposed.

5 SPEAKER-PHONETIC VECTOR REPRESENTATION FOR SHORT DURATION UTTERANCE

In Chapter 4, the generalised variability model has been proposed to obtain a single vector to represent one utterance. In this chapter, this single vector representation will be shown to be sensitive to phonetic mismatch in short duration utterances. In order to address this problem, three methods to obtain vectors that contain speaker-specific and phonetic information are proposed in this chapter. The basic assumption of these methods is that speaker discriminative information in an utterance is delivered with different phonemes and different phonemes may contain different aspects of speaker discriminative information. This is different with the idea of [16], whereby phonetic information is intended to be removed by adding a factor in total variability model. But the results of this linear model do not support that the phonetic information can be efficiently removed. The proposed methods use the idea that the representation of an utterance will be 'softly' (with probabilities as opposed to hard decisions) divided into a number of vectors which have speaker and phonetic class meaning. The phonetic class is either a group of phonemes or a single phoneme. Consequently, models using this idea can compare speaker discriminative information within each phonetic class and thus have some inherent content matching ability that is beneficial for short duration ASV mentioned in Section 2.4. This is achieved by three methods. The first one is the revised GVM to have speaker-phonetic vector representations. The second model is mixture of total variability model. The third one is the tied the parameters of mixture of total variability model. These models are all generalised models, which have potential uses in other research areas.

5.1 Phonetic variability in the i-vector space for short utterances

The i-vector/GPLDA system serves as the state-of-the-art in short duration speaker verification. As mentioned in Section 2.4, long duration utterances are reasonably expected to cover all of the acoustic events (e.g., phonemes) and each one has sufficient information. The contents of utterances are normalised by later transformations and representations. However, these are properties that are helpful in ASV system but not met in short duration utterance. As discussed in Section 2.4, this is due to several reasons. Firstly, in short utterance, one utterance may not cover all the acoustic events. That means there are some 'acoustic holes' in the short duration utterances [85]. Secondly, the relative amount of speaker discriminative information in each acoustic event is more diverse in short duration utterances compared with long duration utterances. How those acoustic events are visited in short duration utterances has strong influence on utterance representation and potentially makes it sensitive with contents. Besides reasons, the main point here is that i-vectors estimated from long utterances are more likely to be clustered together because the statistics of different phonemes are stable, while those of short duration utterances are not. i-vectors of different phonemes have a tendency to cluster, which contributes to the fact that i-vectors from short duration have larger within-class variation.

To support the claim above, further analysis of the phonetic variability in i-vector space is presented in this section, additional to the analysis of the total variability model presented in Section 4.1. i-vectors of different phonemes from different utterances are collected together using a phoneme decoder. Frames that are recognised as the same phonetic class for a given utterance are grouped together to estimate the corresponding i-vector. Those i-vectors are then projected into a two-dimensional space by principle component analysis (PCA).

Figure 5.1 shows the result of this analysis. 304 utterances randomly selected from background databases, including NIST SRE'04, 05, 06, 08, Switchboard II Part 1, 2, 3 and

Switchboard Cellular Parts 1 and 2, are used. It is clear that different groups of phonemes tend to cluster together. This indicates that i-vectors from different phonemes are not uniformly distributed in the original i-vector space and supports the ideas that an i-vector is not phonetically invariant and that the output of the total variability model contains phonetic information. This would not be a problem for long duration utterances in the total variability model as the amount of information is sufficient to cover all phonetic events and the statistical patterns for each group are relatively stable. Consequently, the extracted i-vector will not be biased toward a particular group and the within-class covariance is not enlarged. However, for short durations, the amount of information in each group is not statistically stable. This will make the extracted i-vector biased to some dominant groups, and therefore introduce further mismatch to scenarios where long duration utterances are served as enrolment data and short duration utterances are used as test files. This is the major reason that i-vectors is sensitive to content variation in short duration utterances.

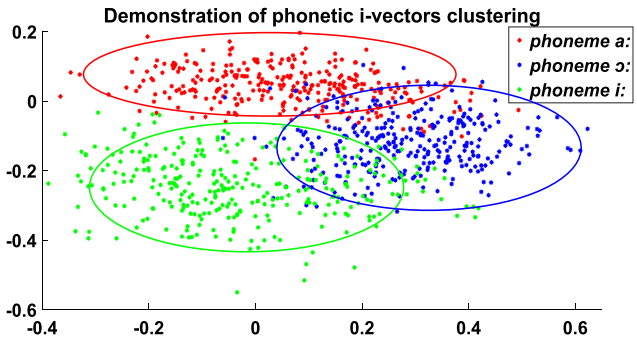


Figure 5.1 Demonstration of phonetic i-vector clustering in two-dimension space.

Together with the analysis of total variability model as in Section 4.1, it can be seen that the total variability model has many problems when it comes to short duration utterances; especially with regards to phonetic variation, different modelling techniques should be proposed for short utterances. In Section 5.2, revised generalised variability model is proposed to have vector representations of utterance with speaker- and phonetic meaning. In Section 5.3, mixtures of

total variability models are proposed to model short duration utterances. Phonetic meaning is ascribed to those vectors and this endows those vectors with content matching ability. Since amount of information in each phonetic class, indicated by the number it has been visited, is different, regarding them as reliable as the same is not a reasonable assumption. Thus, reliabilities, represented by statistics of utterance, of vector phonetic class will be propagated into scoring stage.

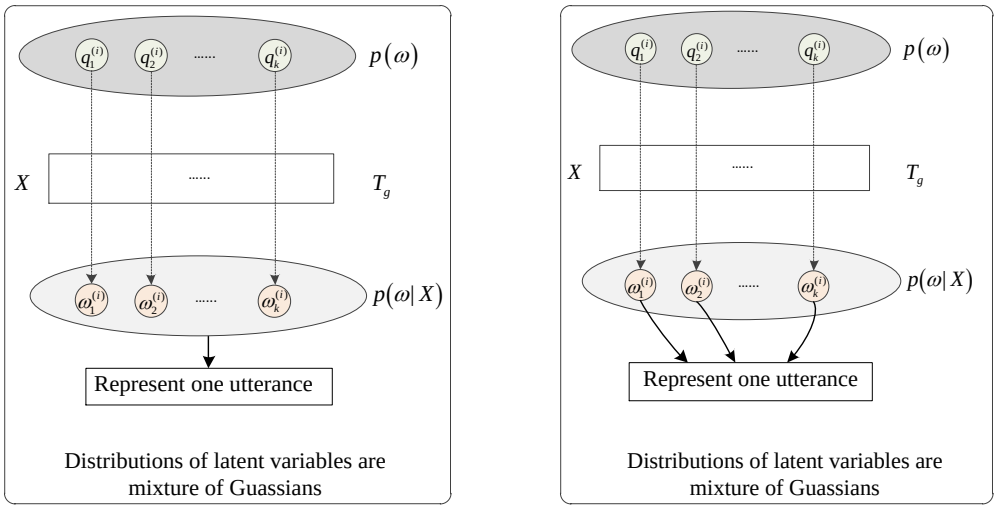
5.2 Revising GVM to generate phonetic-speaker vectors

In Section 4.2, GVM is proposed to generate better representation of utterance compared with TVM. However, one single vector is obtained to represent one utterance. Based on the observations of Section 5.1, which supports the idea that different distributions need to be assigned to different phoneme groups, it may be beneficial to revise the GVM to generate phonetic-speaker vectors to represent one utterance. Let us begin with the assumptions that different phonemes will have different priors, and that the latent variables are generated from different sources. Each source has its own prior and these priors bear the full burden of mapping phonetic information into different groups. The latent variable distribution is a combination of these different groups. The development of this method is similar to that described in Section 4.3, with the main difference being that that instead of using the expectation of the latent variable given observed data $E[\omega|X]$ (4-15) in Section 4.2.2, the expectation of the latent variable given observed data and source states $E[\omega|q_k, X]$ is used instead. It and the corresponding covariance is calculated as follows

$$cov(\omega|q_k, X) = \left(\boldsymbol{B_k^{-1}} + \boldsymbol{T}^*\boldsymbol{\Sigma^{-1}}\boldsymbol{NT}\right)^{-1} \tag{5-1}$$

$$E[\omega|q_k, X] = cov(\omega|q_k, X)\left(\boldsymbol{T}^*\boldsymbol{\Sigma^{-1}}F + \boldsymbol{B_k^{-1}}m_k\right) \tag{5-2}$$

The rest of the development of the model parameters is the same as the generalised variability model in Section 4.3 from (4-1)-(4-24).



(a) (b)

Figure 5.2 Difference between (a) GVM, and (b) revised GVM to obtain phonetic-speaker vectors to represent on utterance.

Thus, as presented in Figure 5.2, the difference between the model in this section and GVM is that instead of using $p(\omega|X)$ to represent utterance, $p(\omega|X, q_k)$ is used.

5.3 Proposed mixtures of the total variability model

Prior to the total variability model, the eigenvoice model was proposed [108]. The eigenvoice model is a factor analysis model that laid the mathematical foundation for total variability model. The difference between eigenvoice model and total variability model is that speaker labels are not used in the total variability model. Kenny et al. mentioned that a single Gaussian assumption may not be optimal in the case of the eigenvoice model [108]. A prior specified with a mixture of Gaussians should be beneficial if a large number of speakers are available for training purposes. The mixture of probabilistic principal component analysis (PPCA) method proposed in [113] has independent latent variables for each mixture component. The total variability model is a variation of this model that ties latent variables across all mixture

components. as can be seen from the graphical model in Figure 5.4 which is identical to Figure 2.7 and presented here for ease of comparison. Figure 5.3 illustrates the mixture of total variability model. Note that the curve line in this picture denotes a Gaussian mixture. This is only for illustration purpose. The model proposed here has a hierarchical structure in which the first layer is similar with to mixture of PPCA, but under each mixture component of the first layer, there is a complete total variability model with a shared latent variable. This model is shown in a graphical model in Figure 5.5. It is similar to the latent Dirichlet allocation model [131] in the way that several latent variables are allocated to one utterance or document, and that how to allocate those variables is determined and inferred by the model and observable data.

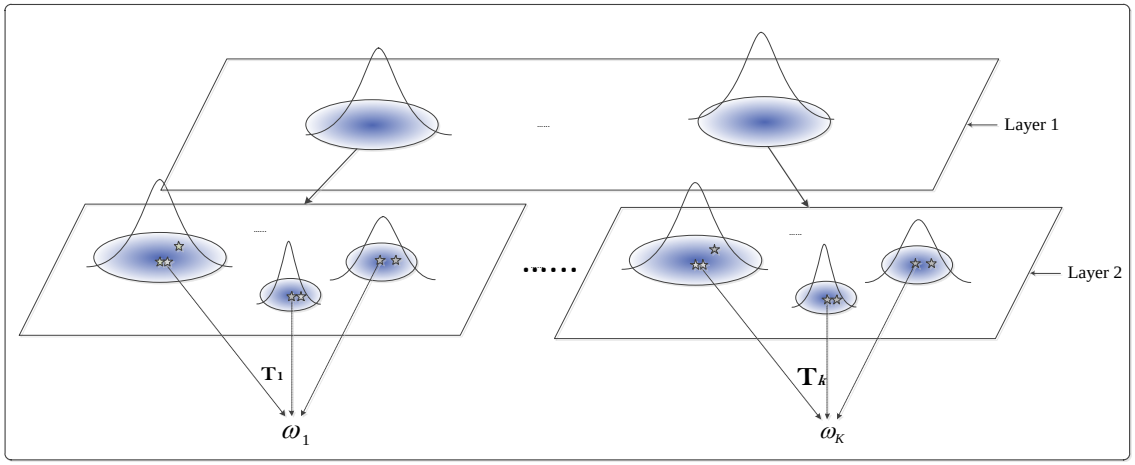


Figure 5.3 Illustration of mixture of total variability model.

In the model presented in Figure 5.5, observed feature frames are first aligned to a phoneme decoder to divide them to different phoneme groups. Under these different groups, feature frames are then again aligned to a GMM dedicated to a particular phonetic group, which is called as phonetic UBM and denoted as pUBM. This procedure is shown in the graphical model by the grouping variable K and labelling variable Z_k. Unlike the conventional total variability model, in which one supervector is derived to represent a given utterance, in the proposed method, there are K supervectors $M_k = [\mu_{1,k}\ \mu_{2,k} \cdots \mu_{C^k,k}]$ such that

$$M_k = M_{0k} + \boldsymbol{T_k}\omega_k \tag{5-3}$$

where the latent variable $\omega_k \sim \mathcal{N}(0, I)$. This means that this model uses *K* supervectors in each phonetic group to represent a utterance. In each phonetic group, there is a total variability model to map information in this phonetic group into a lower rank linear manifold. The alignments of the given observed frames to different phonetic groups are provided by a phonetic decoder, which can be trained before-hand.

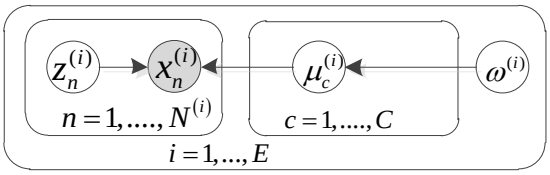


Figure 5.4 Graphical model representation of a total variability model. The variables are: z - labelling variables; x - feature frames; μ - means of the supervectors; ω - latent variable. The indexes are: superscript i – utterance index; subscripts c - mixture component in UBM, and n - feature frame index.

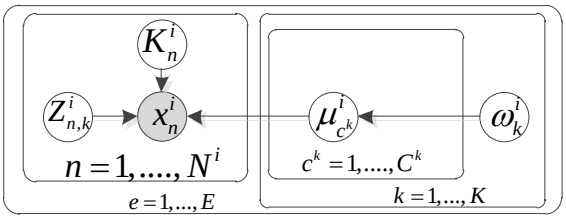


Figure 5.5 Graphical model of mixture of total variability model. The variables are: z - labelling variables; x - feature frames; μ - means of the supervectors; ω - latent variable; K – phonetic class labeling variables. The indexes are: superscript i – utterance index; subscripts c - mixture component in UBM, k – phonetic class index, and n - feature frame index.

To illustrate this in probabilistic way, in the total variability model, one single latent variable is assigned. In the proposed model, each phonetic class will be assigned with a latent variable and a number of phonetic classes are applied. The corresponding probability density function is written as

$$P(O_t) = \sum_s \pi_s \sum_{c \in \mathcal{G}_s} \pi_{c|s} P(x_n | s, c) \tag{5-4}$$

where s denotes the phonetic class and c denotes the mixture component; π_s are the mixing coefficients in the first layer satisfying $\sum_s(\pi_s) = 1$; $\pi_{c|s}$ are the mixing coefficients in the second layer given the set $\mathcal{G}_s$ in the hierarchical structure and satisfying $\sum_c(\pi_{c|s}) = 1$; and $P(x_n|s,c)$ is the probability when the phonetic class and mixture component label in second layer are given, which follows a Gaussian distribution such that $P(x_n|s,c) = \mathcal{N}(x_n|\mu_{c,s}, \boldsymbol{B_{c,s}})$.

This model is derived with the idea that each phonetic class has its own latent variables and information of each class is softly aligned in this model.

Each frame of a given utterance is firstly aligned to different speech classes in the first layer (e.g., phonetic class). In the second layer, zeroth- and first-order statistics are calculated by a phonetic class based UBM. These can be regarded as information in an utterance that has been divided into several phonetic groups. In each group, there is a supervector and a total variability model built on this layer. A lower dimensional representation is then derived in each class to incorporate both speaker and phonetic information. It is expected that within each phonetic group, the phonetic variation of a short utterance is diminished as the results of the phoneme alignment procedure, thus phonetic variation of short utterances is mainly absorbed by information deployment into different groups.

As the number of latent variables is large in this model, it is difficult to obtain the posterior probability by directly using the EM algorithm introduced in Appendix A. A variational Bayesian EM (VBEM) is then used. In this method, the prior of the latent variables are assumed to have a factorised form. Variational posterior probabilities of latent variables, which are derived by minimising the KL divergence which introduced in Appendix A, are calculated to approximate the true posteriors. These variational posterior probabilities are used in the Maximisation step and the log-likelihood is guaranteed to increase. The development of the algorithm follows.

First, let us denote the latent variables as: $z_t^{(s)}$, the phonetic class label variables for s^{th} phonetic class (binary variables); $z_t^{(c|s)}$, the c^{th} mixture component label variables in the s^{th} phonetic class (binary variables); and ω_s, the latent factor for the s^{th} phonetic class. For simplicity, the index indicating the utterance has been omitted. The true posterior is approximated by the factorised form of the posterior, which is

$$Q(H) = \prod_s \left\{ Q_{\omega_s}(\omega_s) \left[\prod_n Q_{z_n^{(s)}}\left(z_n^{(s)}\right) \prod_c Q_{z_n^{(c|s)}}\left(z_n^{(c|s)}\right) \right] \right\} \tag{5-5}$$

where H denotes the collected hidden variables. The log-likelihood of the complete data is calculated first. By omitting the session label, it is expressed as following,

$$log[P(X,H)] = \sum_s \left\{ log[P_\wedge(\omega_s)] + \sum_n \left[log\left[P_\wedge\left(z_n^{(s)}\right)\right] + log\left[P_\wedge\left(z_n^{c|s}\right)\right] \right] \right\} + \sum_s \sum_n log[P(x_n|H)] \tag{5-6}$$

where X represents the collected observable features, $P_\wedge(\cdot)$ represents the prior distribution, and

$$log[P(x_t|H)] = \sum_s \sum_n \sum_c \delta\left(z_n^{(s)}, s\right) \delta\left(z_n^{(c|s)}, c\right) log\left[\mathcal{N}\left(x_n; \mu_{c,s}, \boldsymbol{B_{c,s}}\right)\right] \tag{5-7}$$

where $\delta\left(z_n^{(s)}, s\right)$ is equal to 1 when the binary variable z_n^s indicates class s, and otherwise 0; and similarly for $\delta\left(z_n^{(c|s)}, c\right)$.

5.3.1 Calculate variational posterior probability

In EM algorithm, the posterior probabilities are calculated in E-step. In VBEM, the variational posterior probabilities are computed to approximate the true posterior probabilities since the true ones are too complicated to compute in some scenarios. According to [50], the optimal solution for the j^{th} factor (denoted as Q_j^*) can be written in terms of its logarithms

$$log\left(Q_j^*\right) = E(log[P(X,H])_{\sim Q_j} + d \tag{5-8}$$

where $E(\cdot)_{\sim Q_j}$ means the expectation operator over all the factors except Q_j , and d is a constant. Take the latent variable ω_s as an example:

$$log\left(Q^*_{\omega_s}(\omega_s)\right) = log[P_\Lambda(\omega_s)] + E(log[P(X|H)])_{\sim Q_{\omega_s}} + d\_1 \tag{5-9}$$

where d_1 is a constant that is not related to ω_s. This can be calculated by firstly substituting equation (5-7) into the last term above, such that

$$\langle logP(X|H)\rangle_{\sim Q_{\omega_s}} = \sum\nolimits_n \sum\nolimits_s Q^*_{z_n^s}(z_t^s) Q^*_{z_n^{c|s}}\left(z_n^{c|s}\right) E(log[P(x_n|\omega_s, s, c)])_{\sim Q_{\omega_s}} + d\_1 \tag{5-10}$$

where $Q^*_{z_n^s}(z_n^s)$, and $Q^*_{z_n^{c|s}}\left(z_n^{c|s}\right)$ are the posterior probabilities of $z_n^{(s)}$ and $z_n^{(c|s)}$ respectively. Denoting

$$\gamma_n^{(s,c)} = Q^*_{z_n^s}(z_t^s) Q^*_{z_n^{c|s}}\left(z_n^{c|s}\right). \tag{5-11}$$

This means that the posterior probability of phonetic class has been propagated into zero-order statistics calculation.

Next, in (5-10), expanding $log[P(x_n|\omega_s, s, c)]$ results in a number of terms of ω_s. Combine those terms which include variables ω_s and use the stacked forms of the zeroth- and first-order statistics N_s and F_s, and parameters B_{ω_s}as described in [108],

$$\begin{aligned} log\left[Q^*_{\omega_s}(\omega_s)\right] = {} & \omega_s^* \boldsymbol{B}_{\boldsymbol{\omega_s}}^{-\mathbf{1}} \mu_{\omega_s} - \frac{1}{2}\omega_s^* \boldsymbol{B}_{\boldsymbol{\omega_s}}^{-\mathbf{1}} \omega_s + (\boldsymbol{T_s}\omega_s)^* \boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}} F_s \\ & - \frac{1}{2}\boldsymbol{N_s}(\boldsymbol{T_s}\omega_s)^* \boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}} (\boldsymbol{T_s}\omega_s) + d\_2 \end{aligned} \tag{5-12}$$

where, specifically, d_2 is a constant that is different with d_1, F_s is the stacked form of $F_{s,c}$ over the component index c, N_s is the stacked form of $N_{s,c}$, and finally $F_{s,c} = \sum_n(\gamma_n^{s,c} x_n)$ and $N_{s,c} = \sum_n(\gamma_n^{s,c})$.

The posterior distribution still follows a Gaussian distribution. Thus, through some algebraic manipulations, the mean and covariance of the variational posterior are termed as

$$\mu'_{\omega_s} = \left(\boldsymbol{B}_{\boldsymbol{\omega_s}}^{-\mathbf{1}} + \boldsymbol{T}_{\boldsymbol{s}}^{*}\boldsymbol{N}_{\boldsymbol{s}}\boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}}\boldsymbol{T}_{\boldsymbol{s}}\right)^{-1}\left\{\boldsymbol{B}_{\boldsymbol{\omega_s}}^{-\mathbf{1}}\mu_{\omega_s} + \boldsymbol{T}_{\boldsymbol{s}}^{*}\boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}}F_s\right\},$$
$$\boldsymbol{B}'_{\boldsymbol{\omega_s}} = \left(\boldsymbol{B}_{\boldsymbol{\omega_s}}^{-\mathbf{1}} + \boldsymbol{T}_{\boldsymbol{s}}^{*}\boldsymbol{N}_{\boldsymbol{s}}\boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}}\boldsymbol{T}_{\boldsymbol{s}}\right)^{-1}. \tag{5-13}$$

5.3.2 Calculating a lower bound for VBEM

A lower bound of the variational Bayesian EM algorithm serves as an approximation of the log-likelihood. Thus, it is used as the objective function. According to [50], the lower bound is,

$$\mathcal{L}(X, Q^*) = \langle log[P(X, \Omega, S, C)]\rangle_{Q^*} - KL\big(Q(\Omega)||P_\wedge(\Omega)\big) - KL\big(Q(r_n^s)||P_\wedge(r_n^s)\big) - KL\left(Q\left(r_n^{c|s}||P_\wedge\left(r_n^{c|s}\right)\right)\right) \tag{5-14}$$

where $KL(\cdot)$ denotes the KL divergence. As terms including the KL divergence have no relation to parameters in this model, we can view them as constant regarded to optimise against the parameters. Thus, by expanding the first term of (5-14),

$$\langle log[P(X|\Omega, S, C)]\rangle_{Q^*} = \sum_s \left\{\sum_n \sum_c \delta(r_n^s, s)\delta\left(r_n^{c|s}, c\right)\left[-\frac{1}{2}(x_n - \mu_{cs})^*\boldsymbol{B}_{\boldsymbol{cs}}^{-\mathbf{1}}(x_n - \mu_{cs}) + (\boldsymbol{T}_{\boldsymbol{cs}}\omega_s)^*\boldsymbol{B}_{\boldsymbol{cs}}^{-\mathbf{1}} - \frac{1}{2}(\boldsymbol{T}_{\boldsymbol{cs}}\omega_s)^*\boldsymbol{B}_{\boldsymbol{cs}}^{-\mathbf{1}}(\boldsymbol{T}_{\boldsymbol{cs}}\omega_s)\right]\right\} + d\_1. \tag{5-15}$$

Equation (5-14) can be written as:

$$\tilde{\mathcal{L}}(X, Q^*) = \sum\nolimits_s \mathcal{L}_s + d\_2 \tag{5-16}$$

where

$$\mathcal{L}_s = \sum\nolimits_s \left\{(\boldsymbol{T}_{\boldsymbol{s}}\omega_s)^*\boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}}\boldsymbol{T}_{\boldsymbol{s}}F_s - \frac{1}{2}tr\left[\boldsymbol{N}_{\boldsymbol{s}}\boldsymbol{T}_{\boldsymbol{s}}^{*}\boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}}\boldsymbol{T}_{\boldsymbol{s}}E(\omega_s\omega_s^*)\right]\right\}. \tag{5-17}$$

This means that the lower bound is a linear combination of lower bounds in each group $\mathcal{L}_s$.

5.3.3 Parameter update formula

$\mathcal{L}(X, Q^*)$ from equation (5-16) is a linear combination of sub-classes. Including session labels i, and setting the derivative with respect to $\boldsymbol{T_s}$ to zero,

$$\frac{\partial \mathcal{L}(X, Q^*)}{\partial \boldsymbol{V_s}} = \frac{\partial \mathcal{L}_s}{\partial \boldsymbol{V_s}} = \sum_i \left[F_s^i E\left(\omega_s^{i*}\right) - \boldsymbol{N_s^i T_s} E\left(\omega_s^i \omega_s^{i*}\right) \right] = 0. \tag{5-18}$$

Thus equation (5-18) can be simplified to

$$T_s = \left[\sum_i F_s^i \, E\left(\omega_s^{i*}\right) \right] \left[\sum_i \boldsymbol{N_s^i} E\left(\omega_s^i \omega_s^{i*}\right) \right]^{-1}. \tag{5-19}$$

It can be seen that the format of the updating formula is similar with the parameter update formula of total variability model [108] with the difference that zeroth- and first-order statistics are combinations of two layers according to (5-11).

5.3.4 Tying parameters in mixture of total variability model

Inspired by the success of a subspace GMM [132], in which the same projection matrix is shared by all phonetic states, in this subspace GMM system, the different phonetic classes can have different supervectors while still sharing the same total variability model that is trained by considering all phoneme's supervectors.

In the method proposed in this section, supervectors are calculated differently for each phonetic group in each utterance, which means that instead of the zero- and first-order statistics N and F, we instead have statistics N_s and F_s (where $s = 1{:}S$ where S is the number of phonetic groups) will be calculated. The zero- and first-order statistics are calculated in a similar way to that described in Section 4.3.1, with the slight difference that the pUBMs are replaced by the same UBM. Those supervectors will share the same total variability matrix like [132] to project them to a low dimensional space.

The motivation behind tying parameter is that it reduces the computational load by tying the factor loadings in MTVM, but it is still able to represent utterance by a number of vectors with both speaker-specific and phonetic information. The same development schema of MTVM can be applied to this tied parameter model, thus the log-likelihood of the data is guaranteed to increase with the E- and M- steps while tying different factor loading matrices together in the development of mixture of total variability models.

Specifically, the mean and covariance of the latent variables ω_s have the same format with (5-13), with the difference that same factor loading $\boldsymbol{T}$ is shared by all phonemes.

$$\mu'_{\omega_s} = \left(\boldsymbol{B}_{\boldsymbol{\omega_s}}^{-\mathbf{1}} + \boldsymbol{T}^{*}\boldsymbol{N_s}\boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}}\boldsymbol{T}\right)^{-1}\left\{\boldsymbol{B}_{\boldsymbol{\omega_s}}^{-\mathbf{1}}\mu_{\omega_s} + \boldsymbol{T}^{*}\boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}}F_s\right\} \tag{5-20}$$

$$B'_{\omega_s} = \left(\boldsymbol{B}_{\boldsymbol{\omega_s}}^{-\mathbf{1}} + \boldsymbol{T}^{*}\boldsymbol{N_s}\boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}}\boldsymbol{T}\right)^{-1}$$

where $\boldsymbol{T}$ is the factor loading that shared by all phonetic classes. The lower bound in this tied parameter model is written as:

$$\tilde{\mathcal{L}}(X, Q^{*}) = \sum\nolimits_{s} \tilde{\mathcal{L}}_s + d\_2 \tag{5-21}$$

where

$$\tilde{\mathcal{L}}_s = \sum\nolimits_{s}\left\{(\boldsymbol{T}\omega_s)^{*}\boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}}\boldsymbol{T}F_s - \frac{1}{2}tr\left[N_s\boldsymbol{T}^{*}\boldsymbol{B}_{\boldsymbol{s}}^{-\mathbf{1}}\boldsymbol{T}E(\omega_s\omega_s^{*})\right]\right\}. \tag{5-22}$$

In the parameter updating formula, since all the $\boldsymbol{T_s}$ are the same $\boldsymbol{T}$, the derivative of lower bound (5-20) with respect to $\boldsymbol{T}$ is

$$\frac{\partial\tilde{\mathcal{L}}(X, Q^{*})}{\partial V} = \frac{\partial\sum_s\{\tilde{\mathcal{L}}_s\}}{\partial V} = \sum\nolimits_{s}\left\{\sum\nolimits_{i}\left[F_s^i E\left(\omega_s^{i*}\right) - N_s^i\boldsymbol{T}E\left(\omega_s^i\omega_s^{i*}\right)\right]\right\}. \tag{5-23}$$

Thus, by setting this derivative to zero as was done in equation (5-18), the updating formula is

$$\boldsymbol{T} = \left[\sum_s \sum_i F_s^i E\left(\omega_s^{i*}\right)\right]\left[\sum_s \sum_i \boldsymbol{N_s^i} E\left(\omega_s^i \omega_s^{i*}\right)\right]^{-1}. \tag{5-24}$$

It can be seen in (5-24) zero- and first-order statistics of all phonetic groups are used for updating the parameters. This model reduces the computational load compared to (5-19). The efficacy of this method will be tested in experimental Section 5.4.

5.3.5 Scoring method

In these methods, the posterior probability of the latent variable will be a mixture of Gaussians, which means that one utterance can be represented by a number of vectors based on the proposed model. A bank of GPLDAs is then estimated to obtain scores for each phonetic vector. The final score is then calculated by

$$Score(X_e, X_t) = \sum_k \gamma_k Score(X_{ek}, X_{tk}) \tag{5-25}$$

which is similar with (4-40) in Section 4.3.4.

5.3.6 Discussion of mixtures of total variability model

In the total variability model, one utterance is represented by one i-vector. In this chapter, vector representations that have phonetic- and speaker-information are proposed for each utterance. In MTVM, a number of supervectors (depends on how many phonetic classes) will be first formed. Phonetic information is expected to be divided into different classes and absorbed in supervectors. Different classes are assumed to have different distributions, thus will have different parameters to model them, including phonetic UBMs and phonetic T matrices. Those parameters then realised the idea that different phonetic supervector groups are assumed to have different distributions. It should be noted that the proposed methods do not have to confine to phonetic meaning. It can be generalized to any classes.

5.4 Experimental evaluation of phonetic-speaker vector representation

To assess the performance of the proposed methods in this chapter, speaker verification experiments were conducted on two tasks of the NIST SRE 2010 [13] challenge: CORE-10SEC, and 8CONV-10SEC and additional 5 seconds and 3 seconds conditions similar with Section 4.2.4. The baseline system is an i-vector/GPLDA system, the same as in Sections 4.3 and 5.2.

For the proposed methods, standard MFCC features of 13 dimensions with their deltas and delta-deltas which consists of 39 coefficients were used. Identical background datasets are used. The BUT group's phoneme decoder [129] is used to obtain phonetic posterior probabilities in this section. To further simplify the system, similar phonemes (e.g., long and short duration phonemes) are clustered together, resulting in 14 phonetic groups. The phonetic group configuration is the same as in Section 3.2.5. The corresponding phonetic posterior probabilities in the same group are then added up. In MTVM, the phonetic UBMs used in the mixtures of the total variability model were trained by starting from the conventional UBM used in the baseline system and retrained with features belonging to corresponding phonemes that were recognised by the phoneme decoder if the phonetic posterior probability surpassed a given threshold (e.g., 0.5). When retraining the phonetic UBMs, mixtures with small weight values were deleted, resulting in different numbers of mixture components in each phonetic UBM. In the tied parameter version of MTVM, the rank of the total variability matrix is set as 400. The same UBM was used as in the baseline. The proposed methods produced 14 phonetic speaker vectors to represent one utterance. Parameters are trained and updated according to equation (5-19) for MTVM, and (5-24) for Tied MTVM. For the MoG prior system, the procedure of parameter estimation is the same as that found in Section 4.4.5. A bank of 14 parallel GPLDA models is

then trained to obtain verification scores within each group and the weighted sum from equation (5-25) was used to produce the final scores.

Tables 5.1-4 summarise the results of the proposed systems in terms of EER and MinDCF. The system of MTVM is denoted as MTVM, Tied MTVM as 'Tied', and the revised GVM system is denoted as 'MoG Prior'. From these tables, it can be observed that the proposed methods perform better when there is duration mismatch between enrolment utterances and test utterances. All the proposed methods outperformed the baseline system at both CORE-3SEC and 8CONV-3SEC conditions, where an 18.2% relative improvement is observed in the male condition. This supports the argument made in Section 4.1 that the i-vector framework adds additional mismatch to the long enrolment and short test utterance situation and that the proposed speaker-phonetic vectors are able to relieve this mismatch. When the mismatch is not severe, as in CORE-10SEC and 8CONV-10SEC, the proposed speaker-phonetic vector representation is less efficient than the conventional i-vector representation.

Comparing MTVM with tied parameter MTVM, the results of MTVM give consistently better results for 8 CONV-10SEC and additional 8 CONV-5SEC and 8 CONV-3SEC in both male and female conditions, but tied parameter MTVM outperforms MTVM in CORE-10SEC conditions. In the MoG Prior system, the results are generally better than other two methods, especially for the female condition. This seems to suggest that by using different priors for different phonemes, the phonetic information can be mapped to different groups. Phonetic information can then be normalised by comparing phonetic- and speaker-informed vectors. This is a soft version content match, which is efficient for short duration utterances.

Given that the proposed speaker phonetic vectors have the meaning of phonetic class information, which is then expected to complement the total variability framework of the baseline i-vector system (including all phonetic class into one single i-vector), the baseline and the proposed system can similarly be expected to be complementary and fuse well. Thus, the

speaker-phonetic vector systems are fused with the baseline i-vector system at the score level using the BOSARIS Toolkit [130] and denoted as Fusion systems. Fusion1, Fusion2 and Fusion3 in Table 5.1-4 correspond to the fused systems that with MTVM, Tied and MoG Prior, respectively. The NIST SRE 2008 short2-10SEC condition is selected to train parameters for fusion purposes. The Bosaris toolkit [133] is again used for score fusion. Table 5.2 also summarises the fusion results of the proposed systems with the baseline system in score level. It can be seen that substantial improvements are observed for the proposed methods on all conditions and measures. It can also be seen that for shorter conditions, larger improvements are obtained. Relative improvements of up to 17.36% and 14.10% are observed for male and female conditions respectively. This supports the idea that in shorter utterance, phonetic mismatch is more severe, and that the proposed methods mitigate this problem. We can also compare the results with Section 4.4 in the CORE-10SEC condition, the fusion of MoG prior system in male condition in this chapter outperformed the GVM in Section 4.4 while is inferior for female speech. When compared with Table 4.5, it can be seen that comparative or better results can be obtained by methods proposed in this Section.

Table 5.1 EER (in %) results of speaker-phonetic vector representation on the male parts of the NIST SRE 2010 database

EER						
	CORE-10SEC	CORE-5SEC	CORE-3SEC	8CONV -10SEC	8CONV -5SEC	8CONV-3SEC
TVM	8.04	18.45	21.72	5.03	10.73	17.68
MTVM	8.90	14.30	18.92	6.39	11.25	17.34
Tied	9.99	15.02	19.37	6.62	10.76	16.15
MoG Prior	7.38	13.67	15.37	6.87	12.91	16.99
Fusion1	7.16	12.00	17.88	4.36	8.64	14.61
Fusion2	7.79	13.70	19.09	4.83	8.52	14.81
Fusion3	7.04	12.02	16.18	4.52	8.96	14.90

Table 5.2 MinDCF (in %) results of speaker-phonetic vector representation on the male parts of the NIST SRE 2010 database

MinDCF						
	CORE-10SEC	CORE-5SEC	CORE-3SEC	8CONV -10SEC	8CONV -5SEC	8CONV-3SEC
TVM	26.23	51.76	60.65	15.68	29.27	51.30
MTVM	29.09	45.80	58.96	16.74	21.27	54.00
Tied	29.20	46.67	61.79	19.83	32.85	48.64
MoG Prior	22.20	38.67	45.69	21.12	35.48	52.07
Fusion1	20.21	37.32	49.44	12.04	26.71	42.54
Fusion2	21.57	35.18	45.94	12.24	27.51	44.67
Fusion3	21.57	35.18	45.94	12.24	27.51	44.67

Table 5.3 EER (in %) Results of speaker-phonetic vector representation on NIST SRE 2010 of female part

EER						
Female						
	CORE-10SEC	CORE-5SEC	CORE-3SEC	8CONV -10SEC	8CONV -5SEC	8CONV-3SEC
TVM	9.29	17.64	23.14	6.16	12.94	20.85
MTVM	10.73	15.73	21.58	9.01	13.76	20.25
Tied	11.18	16.82	22.63	8.96	13.18	19.32
MoG Prior	10.97	16.31	21.16	6.94	12.05	18.04
Fusion1	8.40	15.40	19.72	5.88	12.42	18.29
Fusion2	8.89	15.52	20.94	6.62	12.09	17.91
Fusion3	8.69	13.91	18.38	5.17	9.90	14.95

Table 5.4 MinDCF (in %) results of speaker-phonetic vector representation on NIST SRE 2010 of female part

MinDCF						
	CORE-10SEC	CORE-5SEC	CORE-3SEC	8CONV -10SEC	8CONV -5SEC	8CONV-3SEC
TVM	29.09	50.73	65.08	18.42	39.84	57.57
MTVM	31.68	50.24	65.58	25.69	41.57	56.38
Tied	33.62	49.15	59.07	25.65	43.98	53.86
MoG Prior	32.25	47.75	60.50	21.59	36.79	55.26
Fusion1	25.64	43.79	59.22	18.04	36.17	52.66
Fusion2	27.19	46.20	61.16	18.70	35.68	52.62
Fusion3	25.11	42.70	55.76	13.84	30.58	44.97

5.5 Summary

In this chapter, it is found that i-vector representation of different phonemes tend to have different distributions, which make the representation sensitive to phonetic mismatch. This mismatch is more severe in shorter utterances. In order to mitigate this problem, speaker-phonetic vector representations based on the revised generalised variability model, the mixture of total variability models (MTVM) and its subsequent tied parameter mixture of total variability models (tied MTVM) are proposed. The basic idea of these methods is to represent one utterance with vector representations that contain both speaker and phonetic information. In the speaker-phonetic vector representation of the revised generalised variability model, phonetic and speaker information is mapped to the posterior probability by different Gaussian mixtures in prior distribution. In MTVM, this information is delivered via a phoneme decoder and total

variability model in each phonetic group. These methods are tested on the NIST SRE 2010 CORE-10SEC and 8CONV-10SEC and additional 5 seconds and 3 seconds conditions. Results show that improvements can be obtained with those methods. Significant improvements are also observed when fusing the baseline i-vector system with the proposed models, suggesting that capturing local phonetic information is complementary. In addition to this, these proposed models are generalised methods that will be beneficial for other problems where other types of information need to be divided in a second level. After proposing better utterance representation for short duration utterances, duration mismatch in utterance representation spaces will be examined in the next chapter.

6 DURATION MISMATCH IN UTTERANCE VECTOR REPRESENTATION SPACE

In Chapters 4 and 5, vector representations including the i-vector, i_g-vector and speaker-phonetic vectors are discussed. These fixed and low-dimensional vectors efficiently represent their corresponding utterances. They also have many advantages mentioned in Section 2.3.3. For example, these vectors occupy a simpler Euclidean space, where techniques like within-class covariance normalisation (WCCN) [60] and linear discriminant analysis (LDA) [61] can be directly applied. Additionally, back-ends such as probabilistic linear discriminant analysis (PLDA) [63] can be used to generate scores in a recognition task.

However, nuisance information such as channel information is still present in these vector representation spaces. Similar to probabilistic principal component analysis (PPCA) [50], vectors generated by the total variability, mixture of total variability and generalized variability models serve to map high dimensional supervectors to a low-dimensional space with little loss in accuracy [108]. There are no inherent mechanisms to remove nuisance information, such as channel information. Channel compensation is then normally applied on top of those vector representations.

In this chapter, it is demonstrated that other than nuisance information like channel information, the duration of utterances is also a factor that causes mismatch between vector representations of enrolment and test data in short duration speaker verification. A model of the

duration mismatch problem in vector representation spaces is formulated in this chapter and addressed by proposing explicit duration mismatch compensation techniques.

6.1 Analysis of duration mismatch in the vector representation space

The total variability model and other methods proposed in previous chapters allow utterances of different durations to be represented by a vector of fixed dimensionality. These vectors are then used to represent corresponding distributions of given utterances. Channel variability is intended to be removed by LDA and GPLDA. However, in the case of short duration ASV, the models of vector representations lead to the vectors reflecting duration variability as well, which in turn affects speaker verification accuracy.

Taking the total variability model as an example, the generative model is given by (2-20) in Section 2.3.3. The information in the feature space is propagated into i-vector space by the posterior probabilities of latent variables, which are given by the equation (2-23) and repeated here again

$$cov(\omega) = (\boldsymbol{I} + \boldsymbol{T}^{*}\boldsymbol{\Sigma}^{-1}\boldsymbol{N}\boldsymbol{T})^{-1} \tag{6-1}$$

$$E[\omega] = cov(\omega)\boldsymbol{T}^{*}\boldsymbol{\Sigma}^{-1}F \tag{6-2}$$

where ω is the latent variable, $\boldsymbol{T}$ is the factor loading matrix, $\boldsymbol{\Sigma}$ is the concatenated covariance of the UBM, and $\boldsymbol{N}$ and F are the stacked forms of the zero- and first-order statistics N_c and F_c, which are calculated as

$$N_c = \sum_n p(x_n|c,\lambda) \tag{6-3}$$

$$F_c = \sum_n p(x_n|c,\lambda)(x_n - \mu_c) \tag{6-4}$$

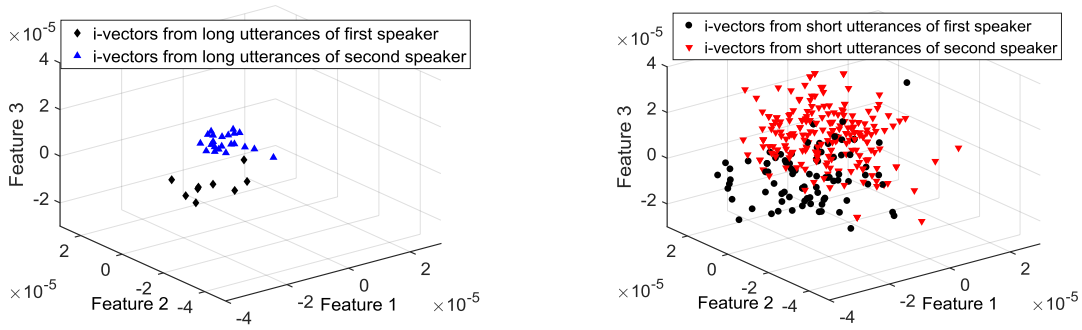
where n denotes the feature frame index, c denotes the mixture in UBM, and λ represents the parameters of the UBM. $\boldsymbol{N}$ then is a $CD_f \times CD_f$ block diagonal matrix whose diagonal blocks are $N_1 I_f, \ldots, N_C I_f$, I_f denotes the $D_f \times D_f$ identity matrix, D_f is the dimension of observed data. $F = [F_1, F_2, \ldots, F_C]$ and C is the total number of mixtures in the UBM.

It can be seen that information in one utterance is propagated into i-vector space as per (6-1) and (6-2). As mentioned in Section 4.1, for long utterances, the amount of information that is represented by zero- and first-order statistics in each mixture is sufficient to represent the utterance. They are also statistically stable, as supported by the results shown in Table 4.1. The first moment $E[\omega|X]$ of the distribution of the latent variable is more representative to the posterior distribution than the one of short duration utterance. However, as the duration of an utterance decreases, the value N_c also decreases because short utterances have less feature frames. Consequently, the uncertainty of the inferred latent variables increases and in turn the i-vectors, which are the posterior means of these latent variables, are affected. Thus, when modelling i-vectors from the same speaker, using the same distribution for i-vectors extracted from both long and short utterances may not be accurate. Firstly, i-vectors extracted from different sessions of short utterance may be expected to have different covariances to the model i-vectors estimated from long utterance for a given speaker. Moreover, due to limited phonetic coverage, the statistics estimated from a short duration utterance are not as representative of the acoustic space as those from a long utterance. This then makes the distribution of i-vectors estimated from short utterances different from that of i-vectors from long utterances for the same speaker.

In order to support the above analysis, i-vectors from different durations are extracted, adopting the same conventional total variability model described in Section 2.3.3 has been adopted. Speakers in NIST SRE'04, 05, 06, 08, Switchboard II Part 1, 2, 3 and Switchboard Cellular Parts 1 and 2 databases are chosen for this demonstration. For each speaker, there are

several sessions which are approximately 2.5 minutes long. These long sessions are used to extract i-vectors for long utterances. Long utterances are then truncated into 10 second utterances, generating a much larger number of utterances. i-vectors are estimated from those truncated 10 second sessions. Linear discriminant analysis (LDA) is then used and trained on the datasets mentioned before. i-vectors which are 400 dimensions from long and short utterances of two different speakers projected onto a three-dimensional space via LDA and showed in Figure 6.1.

From these figures, it can be seen that i-vectors from long utterances are more compactly distributed, leading to more separable i-vector representations. However, those from short utterances are significantly more dispersed, resulting in i-vectors from different speakers that are overlapped and harder to separate. This overlap will significantly degrade the performance of classification.

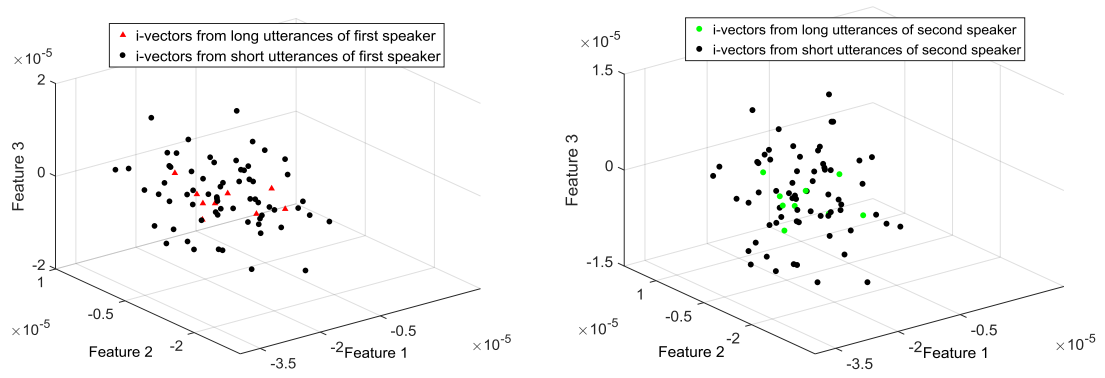


(a) *(b)*

Figure 6.1 Three-dimensional LDA visualization of i-vectors from long and short utterances of different speakers for (a) i-vectors from long utterances, and (b) i-vectors from short utterances.

Moreover, Figure 6.2 shows i-vectors from long and short utterances from two different speakers using a similar projection as in Figure 6.1, but comparing the i-vectors from long and short utterances instead of different speakers. In this figure, black points represent i-vectors from short utterance. Comparing black points with red or green points, it can be seen that black points need larger covariances to model its distribution, which suggests that i-vectors from short

utterances have larger uncertainties and it is expected to have larger covariances to model distributions of i-vectors from short utterance. This means that it is may not be accurate to model i-vectors from long and short utterances with identical distributions, and also signifies the need to normalise the distribution mismatch caused by utterance duration.



(a) *(b)*

Figure 6.2 Three-dimensional LDA visualization of i-vectors from long and short utterances of different speakers for (a) i-vectors from first speaker, and (b) i-vectors from second speaker.

6.2 Duration compensation with linear projection

In Section 2.3.3, the GPLDA model is introduced. It is re-introduced here for the purpose of highlighting how duration mismatch the GPLDA is the lack of the ability to perform duration compensation. The generative equation of GPLDA is (2-11). The latent variables (elements of h_i) are assumed to be statistically independent. The graphical model is repeated here for illustration in Figure 6.3 (a).

However, as the latent variable h_i follows a standard normal distribution and the residual is also distributed normally, the distribution of the summation of the two Gaussian distributed variables is still a Gaussian distribution. More specifically, by marginalising the joint distribution over the latent variables, the i-vectors follow a normal distribution given by $\mathcal{N}(\mu_g, \boldsymbol{\Phi}\boldsymbol{\Phi}^{\boldsymbol{T}} + \boldsymbol{\Sigma}_{\boldsymbol{g}})$. As analysed in Section 6.1, i-vectors estimated from short utterances are

distributed differently to i-vectors extracted from long utterances. This conflicts with the identical distributions assumption in GPLDA. Duration compensation with linear projection before the use of GPLDA is proposed to normalise this distribution mismatch.

6.2.1 Proposed Twin-Model projection method

The aim of duration mismatch compensation is to normalise mismatch between distributions. First, based on the findings of Section 6.1, i-vectors estimated from long and short utterances from the same speaker need to be modelled differently, but should still be linked by speaker identity. As shown in Figure 6.3 (b), the shaded circles represent utterance vector representations of long and short duration, and they are linked by the latent variable h due to the fact that they are spoken by the same speaker.

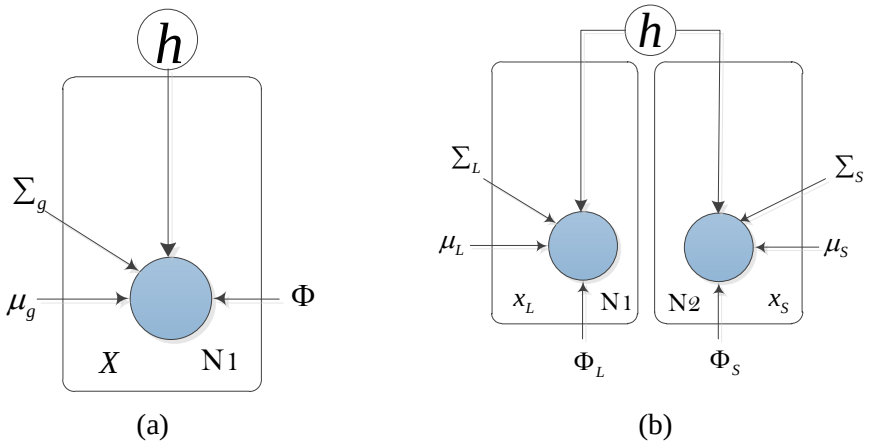


Figure 6.3 Graphical model of (a) standard GPLDA, and (b) Twin-Model projection.

The idea of the projection is that modelling utterance vector representations of long and short duration utterances with two Gaussian distributions and are linked with a single latent variable if they are from the same speaker. Modified from (2-11), the model can be specified by the generative functions

$$\widetilde{\omega}_{ij} = \begin{cases} \mu_L + \boldsymbol{\Phi_L} h_i + \varepsilon_L, & \text{for long utterances} \\ \mu_s + \boldsymbol{\Phi_s} h_i + \varepsilon_s, & \text{for short utterances} \end{cases} \tag{6-5}$$

where $\widetilde{\omega}_{ij}$ denotes the pre-processed i-vector which is introduced in Section 2.4.3; μ_L and μ_s are the mean vectors for the i-vector correspond to long and short utterances, respectively; Φ_L and Φ_s are the corresponding factor loading matrices; and the latent variable h follows $\mathcal{N}(0, \boldsymbol{I})$ and is shared by all utterances i from the same speaker. ε_L and ε_s are residuals which are assumed to follow $\mathcal{N}(0, \boldsymbol{\Sigma_L})$ and $\mathcal{N}(0, \boldsymbol{\Sigma_s})$ respectively. In this model, two factor analysers essentially form a probabilistic based transformation to map the original i-vectors into the same latent variable space that is expected to be duration invariant. The proposed method has the advantages normalising distribution mismatch as well as considering both inter- and intra-speaker variation, which are modelled by the factor loading matrices and covariances respectively.

The generative model is denoted as in equation (6-5) that is same with Twin-Model GPLDA which is a back-end and will be proposed in Section 7.2. The major difference between Twin-Model projection and Twin-Model GPLDA is that the aim of Twin-Model projection is to project vector representations from long and short utterances to a linked latent variable space in order to reduce duration mismatch, and Twin-Model GPLDA is to model vector representations from long and short utterances to generate more reliable scores. The EM algorithm is then used to estimate the parameters, as described in Appendix A. Using the Gaussian assumptions made above and after some algebraic manipulations, the E-step for this model is then formulated as:

$$E[h_i] = \left(I + \sum_k R_i \boldsymbol{\Phi_k^*} \boldsymbol{\Sigma_k^{-1}} \boldsymbol{\Phi_k} \right)^{-1} \sum_k \boldsymbol{\Phi_k^*} \boldsymbol{\Sigma_k^{-1}} \sum_j (\widetilde{\omega}_{kij} - \mu_k) \tag{6-6}$$

$$E[h_i h_i^*] = \left(I + \sum_k R_i \boldsymbol{\Phi_k^*} \boldsymbol{\Sigma_k^{-1}} \boldsymbol{\Phi_k} \right)^{-1} + E[h_i] E[h_i]^* \tag{6-7}$$

where R_i is the number of sessions of i^{th} speaker; $\widetilde{\omega}_{k_{ij}}$ denotes the i-vector corresponding to the j^{th} utterance of the i^{th} speaker, from the k^{th} class (either long, L, or short, S).

In the M-step, the auxiliary function is optimised as

$$Q(\theta, \theta_{old}) = \sum_i \sum_j \sum_k \int p(h_i|X, \theta_{old}) \, log\left[p\left(\widetilde{\omega}_{kij}|h_i\right)p(h_i)\right]dh_i \tag{6-8}$$

where X denotes i-vectors from all training speakers, θ denotes the model hyper-parameters and θ_{old} denotes the model hyper-parameters from the previous iteration of the EM algorithm. By taking the derivatives of Q with respect to θ and setting them to zero, the following update equations are obtained:

$$\mu_k = \frac{1}{N_k} \sum_{i,j} \widetilde{\omega}_{k_{ij}} \tag{6-9}$$

$$\Phi_k = \left(\sum_{i,j} \left(\widetilde{\omega}_{k_{ij}} - \mu_k \right) E[h_i]^T \right) \left(\sum_{i,j} E[h_i h_i{}^*] \right)^{-1}, \tag{6-10}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i,j} \left[\left(\widetilde{\omega}_{k_{ij}} - \mu_k \right) \left(\widetilde{\omega}_{k_{ij}} - \mu_k \right)^* - \Phi_k E[h_i] \left(\widetilde{\omega}_{k_{ij}} - \mu_k \right)^* \right]. \tag{6-11}$$

Note that the parameter estimations are the same with Twin-Model GPLDA which is proposed in Section 7.2. Similar to probabilistic principal component analysis [113], the transformed i-vectors follow a Gaussian distribution and a convenient representation for the posterior mean is found in equation (6-6). To avoid introducing new mismatch into the system, i-vectors of enrolment and test utterances should apply the same transformed matrix. The transformation function is then formulated as

$$\widehat{\omega} = A\widetilde{\omega} - b \tag{6-12}$$

where

$$A = \left(I + \sum_k \Phi_k^* \Sigma_k^{-1} \Phi_k \right)^{-1} \sum_k \Phi_k^* \Sigma_k^{-1} \tag{6-13}$$

and

$$b = \left(I + \sum_k \Phi_k^* \Sigma_k^{-1} \Phi_k \right)^{-1} \sum_k \Phi_k^* \Sigma_k^{-1} \mu_k. \quad (6\text{-}14)$$

6.3 Duration compensation with neural networks

In Section 6.2, a linear projection algorithm was proposed. Unlike the method proposed in Section 6.2, a non-linear projection algorithm implemented with neural networks can be also beneficial. The essential idea is also to normalise the distribution differences within vector representations of long and short duration recordings.

Neural networks have been used in short duration ASV. The major application of neural networks is to map i-vectors from short duration utterances to corresponding ones from long utterances, expecting the mapped i-vectors to be more similarly distributed as i-vectors from long utterances. For example, in [85], a noise model of short duration i-vector was introduced. The idea behind this was that an i-vector estimated from short duration recordings can be regarded as noisy and its corresponding i-vectors for long utterances, called clear i-vectors. By using this model in the GPLDA training phase, and by using a multi-condition training method, the noisy i-vectors are added to the training phase and have proven to be beneficial. A de-noising auto-encoder is also used in speech signal processing. In [134], the autoencoder is used to de-noise the original acoustic feature. In [135], the autoencoders are used to map i-vectors of short duration utterances.

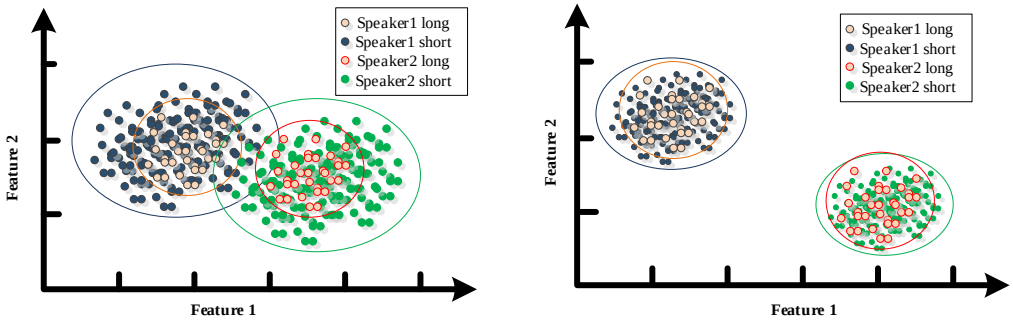
However, the methods mentioned above do not consider how i-vectors from the long utterances transform when training models for short utterances. This means that i-vectors from long utterances are not transformed by those models, which may introduce new mismatch in the situation where enrolment data is long and test files are short utterances. The model proposed in this section is different with the above ones in the way that i-vectors from both long and short

utterances are considered and are mapped into a space that is shared by both long and short utterances. In the centre loss [136], the distances between members of one class and centre of this class are the objectives to minimise. This can be used to reduce the intra-speaker variability in speaker verification. Inspired by this centre loss, a novel objective function is proposed to train a non-linear mapping model in the following section.

6.3.1 Duplet centre loss

As analysed in Section 6.1, variations other than inter- and intra-speaker variability are present in short duration speaker verification. Vector representations from different durations are distributed differently. A projection method that considers all these three variabilities should be used.

To illustrate this idea, Figure 6.4 plots vector representations in two dimensions (note that these figures use simulated data and the made-up two-dimensional distributions is intended for illustrative purposes only). It can be seen that before compensation, vector representations from 'long duration utterances' and 'short duration utterances' are distributed differently, and that vector representations from 'different speakers' are overlapping. After the projection process, two preferred effects can be observed. The first effect is to compensate for variations caused by duration mismatch. This can be measured by the average distance of vector representations of utterances to the centre of corresponding speaker. The other effect is that distributions of different speakers are far enough to make clear verification. This is can be measured by distance between centres of different speakers, and the positive effects are both illustrated in Figure 6.4.b.



(a) *(b)*

Figure 6.4 Illustration of the required compensation effect, showing (a) a 2-dimensional representation of vector representations of utterances from multiple speakers, both long and short utterances, before compensation, and (b) after compensation.

To account for the required effects, a novel loss function is designed. This loss function consists of three components. The first is the distance between vector representations from long utterances to corresponding speaker centres. The second is the distance between vector representations from short utterances to corresponding speaker centres. The final component is the distance between vector representations and centres of all the other speakers. The loss function is formed as:

$$\begin{aligned}\mathcal{L}(\omega_{Ljn}, \omega_{Sjn}|\theta) \\ &= \alpha\left[\tilde{\alpha}D\left(f\left(\omega_{Ljn}|\theta\right) - C_j\right) + \tilde{\beta}D\left(f\left(\omega_{Sj}n|\theta\right) - C_j\right)\right] \\ &- \beta\left[\frac{1}{M-1}\sum_{j\neq i}\left(\tilde{\alpha}D\left(f\left(\omega_{Ljn}|\theta\right) - C_i\right) + \tilde{\beta}D\left(f\left(\omega_{Sjn}|\theta\right) - C_i\right)\right)\right]\end{aligned} \tag{6-15}$$

where ω_{Ljn} denotes the n^{th} i-vector of long utterance from j^{th} speaker, ω_{Sjn} denotes the n^{th} i-vector short utterance from j^{th} speaker, C_j denotes centre of j^{th} speaker, M is the number of speakers in training data, $\sum_{j\neq i}(\cdot)$ is the summation operator that sum over $j\ (1 \leq j \leq M)$ except $j \neq i$, f denotes the non-linear transform, θ denotes the parameters of neural networks,

D denotes a similarity metric (e.g., Euclidean metric or Cosine similarity distance), coefficients $\tilde{\alpha}$ and $\tilde{\beta}$ control the loss contribution from different duration classes, hyper-parameters α and β control the loss contribution from duration and inter-speaker variation, and M is the number of speakers in the training data. As this loss function consists of both vector representations from long and short utterances and centres of speakers, it is called the duplet centre loss function.

If the Euclidean metric is used as the similarity metric, the gradients of this loss function are easy to calculate, and are computed as:

$$\frac{\partial \mathcal{L}(x_{Ljn}, x_{Sjn}|\theta)}{\partial x_{Ljn}} = 2\alpha\tilde{\alpha}(x_{Ljn} - C_j) - 2\frac{\beta\tilde{\alpha}}{M-1}\sum_{j\neq i}(x_{Ljn} - C_i), \tag{6-16}$$

$$\frac{\partial \mathcal{L}(x_{Ljn}, x_{Sjn}|\theta)}{\partial x_{Sjn}} = 2\alpha\tilde{\beta}(x_{Sjn} - C_j) - 2\frac{\beta\tilde{\beta}}{M-1}\sum_{j\neq i}(x_{Sjn} - C_i). \tag{6-17}$$

The parameters can be updated iteration by iteration by using the above derivatives which is listed in Algorithm 6.1.

Similar to the centre loss of [137], the centres are updated each iteration by mapped vector representations. This is accomplished by the following update equation

$$C_i^* = C_i + \lambda\frac{\sum_j^J \delta(S(j) = i)\cdot\left(\tilde{\alpha}/2\cdot\left(f(x_{Lj}|\theta) - C_i\right) + \tilde{\beta}/2\left(f(x_{Sj}|\theta) - C_i\right)\right)}{1 + \sum_j^J \delta(S(j) = i)} \tag{6-18}$$

where C_i^* is the updated centre for i^{th} speaker, $\delta(\text{condition}) = 1$ if the condition is satisfied, otherwise $\delta(\text{condition}) = 0$. λ is in the range [0,1] and is assigned to control the speed of centre updating.

The training algorithm can be described as follows:

Algorithm 6.1. Duration compensation in vector representation space using duplet centre loss.

Input: Training data $\{x_{L11,} \dots, x_{L1J_1}, \dots, x_{LM1,} \dots, x_{LMJ_M}, x_{S11,} \dots, x_{S1J_1}, \dots, x_{SMJ_M,}\}$. Initialize θ and centres C_i. The number of iterations $iter \leftarrow 0$.

Output: θ

1. **while** *iter* **less than** *max_iter* **do**
2. $iter \leftarrow iter + 1$
3. Compute $\mathcal{L}(x_{Ljn}, x_{Sjn}|\theta^{iter})$
4. Compute the backpropagation error $\frac{\partial \mathcal{L}(x_{Ljn}, x_{Sjn}|\theta)}{\partial x_{Ljn}}$ and $\frac{\partial \mathcal{L}(x_{Ljn}, x_{Sjn}|\theta)}{\partial x_{Sjn}}$ for each j
5. Update parameters θ
6. Update centers C_i
7. **End while**

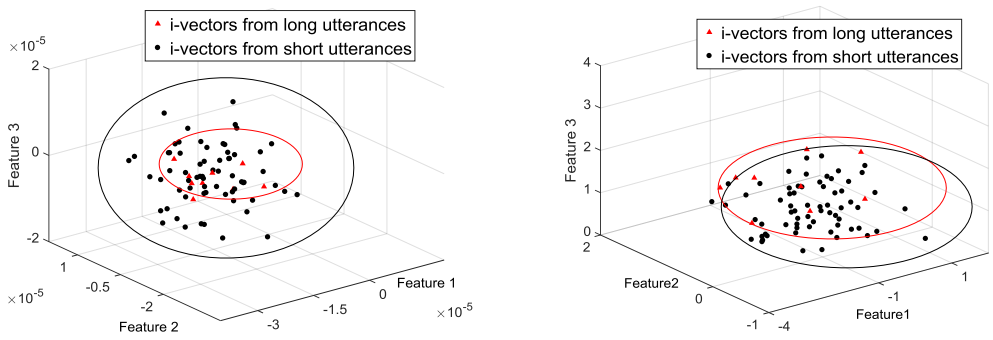
6.4 Experiments

6.4.1 Experiments with Twin-Model projection

The 8CONV-10SEC condition (CC5) of the NIST SRE 2010 dataset was chosen for evaluation. Two additional conditions were created by truncating the 10 second test utterances to 5 and 3 seconds, using the first 5 seconds and 3 seconds of each utterance.

The baseline system is an i-vector/GPLDA system, the same as in Sections 3.3 and 4.2. For the proposed methods, same MFCCs features of 13 dimensions with their first and second derivatives were used. LDA was then applied to further reduce i-vector dimension to 300. Utterances were truncated to 20 seconds and the corresponding i-vectors were estimated to serve as a short duration class to train the proposed method.

Figure 6.5 shows i-vectors of one speaker that are mapped into three dimensions by LDA before and after the proposed transformation. It can be seen that after the transformation, distribution mismatch between the i-vectors from long and short utterances is mitigated. KL divergence [138] is also used to quantify the distribution mismatch. Diagonal Gaussian models were used to model distributions of i-vectors of each speaker from the background data due to the limited data points for each speaker. i-vectors from long and short utterances are modelled separately and the KL divergence is calculated between these two Gaussian models for each speaker. The average values of KL divergence before and after the transformation are 1206 and 520 respectively, which suggests that in the transformed space, the distribution mismatch is lower.



(a) *(b)*

Figure 6.5 Three-dimensional LDA visualization of i-vectors from long and short utterances of the same speakers for (a) i-vectors before transformation (b) i-vectors after transformation.

GPLDA is then applied to model i-vectors after the proposed method. For comparison, the standard GPLDA system is used to directly model i-vectors before the proposed method and the standard i-vector/GPLDA system serves as the baseline system. Before GPLDA modelling, i-vectors were radially Gaussianised, followed by length normalisation as described in [64]. Table 6.1 summarises the performances. The system with proposed Twin-Model projection is denoted as TM-projection. It can be seen that substantial improvements can be obtained when using the proposed method. The proposed method has superior performances in all three conditions

compared to the baseline system. Substantial improvements are also observed for all three conditions, and for both male and female as well. Up to 20% and 29.1% relative improvements are seen for the male and female 3 seconds conditions respectively, using the proposed duration compensation methods. Those improvements support the idea that it is better to normalise the distribution mismatch in the i-vector space prior to scoring.

Table 6.1 Comparison of the performance (EER %) of the standard baseline proposed methods evaluated on the NIST SRE'10 8CONV-10SEC condition as well as the additional 5SEC and 3SEC conditions

	Male			Female		
	Test duration					
	EER%					
	8CONV-10SEC	8CONV-5SEC	8CONV-3SEC	8CONV-10SEC	8CONV-5SEC	8CONV-3SEC
i-vector/ GPLDA	5.12	10.61	17.43	7.01	14.43	20.86
TM-projection	3.58	8.08	13.94	5.71	10.71	14.77
	MinDCF (%)					
i-vector/ GPLDA	14.51	27.43	48.04	18.33	37.31	53.39
TM-projection	11.54	27.15	44.17	17.63	30.83	45.23

6.4.2 Experiments with non-linear projection

To implement Algorithm 6.1, a structure of neural networks is needed. The strength of deep neural networks is that more than one layer can be built to be more efficiently representative and transform the inputs. In this experiment, the structure specified in Figure 6.6 is used. The input is a duplet, which consists of two elements. One is vector representation of a long utterance, and the other is a vector representation of short utterance. The same parameters are shared by both vector representations from the long and short utterances to map them to a new space denoted

by $f(\omega|\theta)$, where ω represents i-vectors. In the new space, the duplet centre loss function (6-16) is applied to train the parameters.

The neural networks used in this experiment are fully connected layers. Two non-linear layers with a rectified linear unit (ReLU) activation function and one linear layer are applied. The numbers of elements in each layer are 1024, 512, 400. The numbers of elements are empirically chosen by the idea of expanding the input and then mapping it to a new compact space. Before being fed into the neural networks, the inputs are L2 normalised. L2 normalisation is also used before applying duplet centre loss.

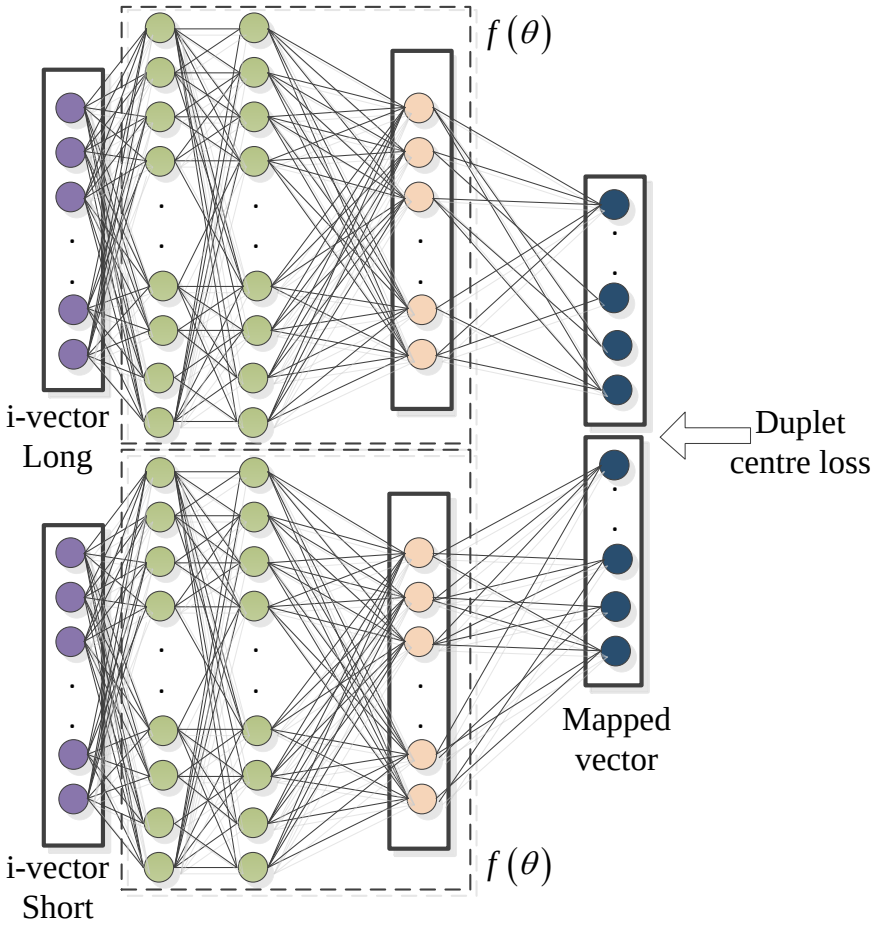


Figure 6.6 Structure of a neural network for duration mismatch compensation.

The 8CONV-10SEC condition (CC5) of the NIST SRE 2010 dataset was chosen for evaluation. As in Sections 4.4, 5.4 and Section 6.2.2, two additional conditions were created by truncating the 10 second test utterances to 5 and 3 seconds, using the first 5 seconds and 3 seconds of each utterance. The i-vector representation is chosen to evaluate this non-linear compensation technique. The same UBMs and T matrices as those found in Section 6.2 are used to extract the i-vectors. The background data and evaluation data are also the same as described in Section 6.2. Long duration utterances were truncated to 20 seconds and the corresponding i-vectors were estimated to serve as a short duration class to train the proposed method. After the non-linear transformation, conventional LDA and GPLDA are both used on top of the compensated i-vectors.

The TensorFlow [139] computation platform is used to implement this algorithm. Male and female conditions are trained and tested separately. The minibatch size is set as 128, and a learning rate is 0.001 is chosen. In order to give a higher weight for compensation of duration mismatch, α is set as 0.75, β is 0.25 and λ is 0.5. The parameters are chosen to optimize evaluation results of 8CONV-10SEC condition (CC5) on the NIST SRE 2010 dataset in a preliminary experiment.

The loss curve is shown in Figure 6.7. During training, it is observed that the losses continuously decrease as the steps increase, but the drop is steeper before 500 steps are completed. This indicates the algorithm converges around 500 steps and settles to the local minimum. The parameters at 500 steps are then used to map i-vectors to a new space.

To gain more insights into how the algorithm performs, Figure 6.8 shows the visualisation of i-vectors from different speakers that are projected into two dimensions by PCA. Numbers from 0 to 9 denotes the 10 speakers. It can be seen that before training, the i-vectors from different speakers are overlapped. After training, i-vectors from the same speakers are more compact and i-vectors from different speakers are separated. This means that intra-speaker variability is

decreased, and inter-speaker variability is increased. It indicates that the algorithm successfully complete two desired effect.

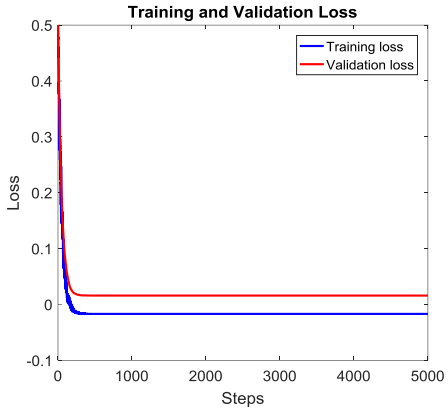
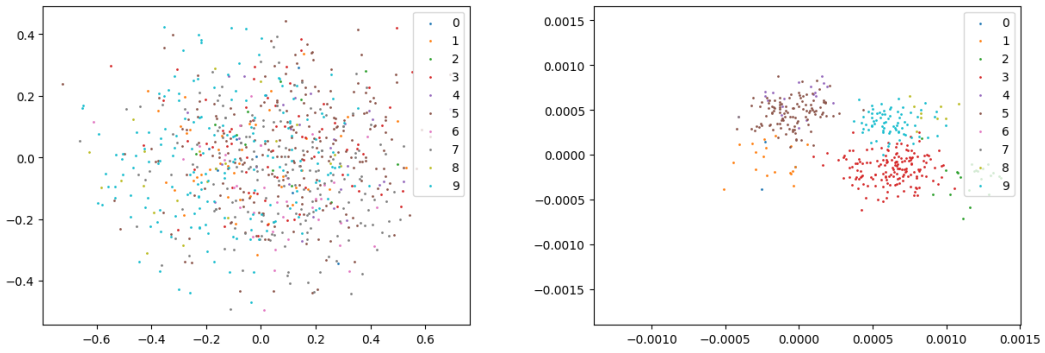


Figure 6.7 Training and validation loss.



(a) (b)

Figure 6.8 Two-dimensional visualisation of vector representations from different speakers (a) before training and (b) after training.

Table 6.2 shows the comparison of the performances of the proposed systems from this section. In this experiment, the conventional i-vector/GPLDA system is the baseline, which is denoted as the No_Compensation system. The system incorporating the non-linear transformation is called the proposed method system. It can be seen that the proposed method outperformed the baseline in five out of six conditions in terms of EER. A relative improvement up to 16.3% was achieved for the 3 seconds female condition. It can also be seen that this algorithm showed higher improvements for shorter duration conditions. The MinDCF metric also confirm the robustness of the algorithm.

In order to show the importance of duplet centre loss, $\alpha = 1.0$, $\beta = 0.0$ are set for comparison. $\beta = 0.0$ means that the loss function forces the loss to only compensate the variability caused by duration, and to ignore the intra-speaker variability. Table 6.3 shows the results of this experiment. It shows that without considering intra-speaker variabilities, merely compensating the duration mismatch actually leads to worse performance. This supports the idea that both variabilities caused by duration mismatch and inter- and intra-speaker need to be considered.

Table 6.2 Comparison of the performance (EER % and MinDCF %) of the standard and proposed methods evaluated on the NIST SRE'10 8CONV-10SEC condition as well as additional 5SEC and 3SEC conditions ($\alpha = 0.75, \beta = 0.25$)

	Male			Female		
	Test duration					
	8CONV-10SEC	8CONV-5SEC	8CONV-3SEC	8CONV-10SEC	8CONV-5SEC	8CONV-3SEC
	EER (%)					
No_Compensation	5.06	10.54	16.72	6.14	12.52	18.80
Proposed method	4.40	8.68	15.83	6.85	12.34	15.74
	MinDCF					
No_Compensation	14.51	27.43	48.04	18.33	37.31	53.39
Proposed method	14.97	27.88	45.05	19.92	34.04	47.58

In comparison, we can see that the proposed linear projection outperformed the non-linear one from the results in Table 6.1 and Table 6.2. The non-linear projection methods may need more data to learn a robust model. Furthermore, the experiments using the non-linear projection were not exhaustive and the structure of DNNs can be further fine-tuned and normalisation techniques such as dropouts can also be applied. This will be pursued as future work.

Table 6.3 Comparison of the performance (in terms of EER % and MinDCF %) of standard , proposed method evaluated on the NIST SRE'10 8CONV-10SEC condition as well as additional 5SEC and 3SEC conditions ($\alpha = 1.0, \beta = 0.0$)

	Male			Female		
	Test duration					
	8CONV-10SEC	8CONV-5SEC	8CONV-3SEC	8CONV-10SEC	8CONV-5SEC	8CONV-3SEC
	EER (%)					
No_Compensation	5.06	10.54	16.72	6.14	12.52	18.80
Proposed method	7.42	12.24	18.80	7.76	13.17	19.07
	MinDCF (%)					
No_Compensation	14.51	27.43	48.04	18.33	37.31	53.39
Proposed method	20.64	41.02	50.64	21.12	36.54	52.94

6.5 Summary

This chapter proposes two methods to normalise the distribution mismatch between long and short duration utterances in the i-vector and similar vector representation spaces. The first proposed transform is a linear transformation that utilises two sets of parameters to model distributions of vector representations from long and short utterances, but the same latent variables are shared for vector representations from the same speaker. Additionally, this transformation maps the utterance vector representations to a latent variable space. Consequently, the proposed method can capture both inter- and intra- speaker variability while compensating for the duration mismatch between long and short utterances. The second proposed transform is a non-linear transformation implemented with neural networks. A novel loss function named the duplet centre loss is proposed. This loss function considers both duration mismatch and inter- and intra-speaker variability, normalising them in one loss function. Experimental results obtained on the NIST SRE 2010 database validate the effectiveness of the proposed transforms. Besides the compensation in utterance representation spaces, the mismatch compensation can also be performed in the back-end, which will be analysed in the next chapter.

7 DURATION MISMATCH COMPENSATION IN THE BACK-END

In previous chapter, compensation techniques for duration mismatch in vector representations are proposed. However, this mismatch may also propagate into back-end classifiers which are typically statistical models, such as Gaussian Probabilistic Linear Discriminative Analysis (GPLDA). Duration mismatch violates the typical assumptions of identical distribution in the back-end. It has been shown that duration mismatch between enrolment and test data can lead to significant degradation in performance [116], and that this mismatch needs to be addressed in order to make short duration speaker verification a viable option.

Some methods to tackle this problem have been previously proposed [85, 115, 116, 121] under the assumption that an i-vector is a reliable representation of a given utterance. For example, in [85], score domain compensation was introduced for duration mismatch. However, given that the i-vector is a point estimate based on the posterior distribution of the latent variables in a total variability model, the uncertainty of this point estimation will become larger as the duration of utterance reduces [108]. Thus, the i-vector becomes less reliable when the utterance is short. A variation of the GPLDA model that takes into account this uncertainty of the i-vector was then proposed and shown to be effective [88, 140] and details of this model will be introduced in Section 7.3.1.

In this chapter, it is shown that, even though pre-processing is performed (including the length normalisation introduced in Section 2.3 and without duration compensation), duration mismatch exists and need to be considered in a back-end classifier. Based on the analyses in Chapter 6, it may be hypothesised that modelling vector representations with a single model may not be optimal. The use of different distributions to model vector representations is

proposed, but these are also connected by the fact that they have the same speaker identity. This is accomplished by tying the same latent variables for these two different distributions. Later, uncertainty propagation is included to take the uncertainty of the vector representations into account.

7.1 Duration mismatch analysis in pre-processed vector space

In Section 6.1 vector representations are raw and have not been pre-processed, while in this section, it is analysed that duration mismatch exists before scoring, even though pre-processing is performed (without duration compensation), and that this needs to be considered in the design of a back-end classifier.

As mentioned in Section 2.3.3, i-vectors are assumed to follow a standard normal distribution after whitening which is to transform input i-vectors to have an identity covariance matrix [64] in GPLDA, and consequently the length of the i-vector should follow a chi-square distribution. Histograms of the L2 magnitude of 200 dimensional i-vectors from long and short utterances are plotted in Figure 7.1. These are obtained from 9,189 i-vectors estimated from the NIST SRE ‘04, ‘05, ‘06, ‘08, Switchboard II Parts 1, 2 and 3, and Switchboard Cellular Parts 1 and 2 full conversation utterances. Correspondingly, the 9,189 i-vectors for short utterances are extracted from utterances by truncating these full utterances and using the first 10 seconds. Note that these i-vectors are transformed by linear discriminative analysis (LDA), within-class covariance normalisation (WCCN) and whitening as GPLDA modelling is built on top of those transformations. From Figure 7.1 it can be seen that the histograms of the length of i-vectors from long and short segments are quite distinct, suggesting that both long and short i-vectors are not identically distributed.

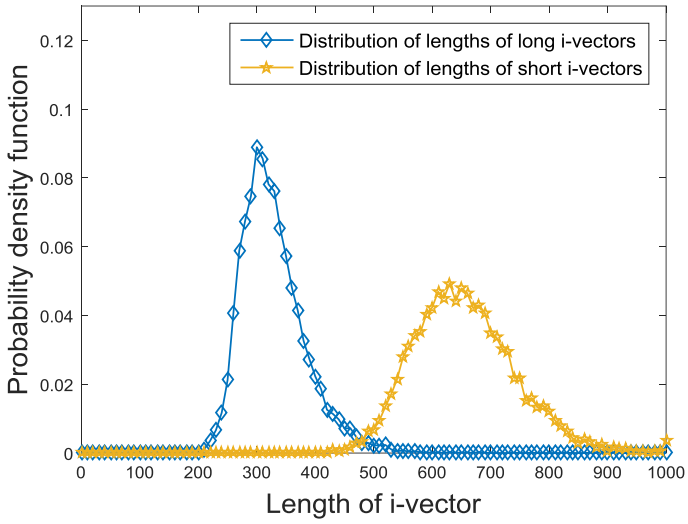


Figure 7.1 Histograms of i-vector lengths (magnitudes) estimated from long and short duration utterances (denoted as long and short i-vectors).

In addition to comparing these histograms, a second measure of differences between the distribution of i-vectors from long and short utterances based on the partition coefficient [117, 141] is employed in this paper. The partition coefficient is an index that indicates the clustering tendency in a dataset and lies in the range $[1/K, 1]$, where K is the number of clusters. A partition coefficient value close to unity indicates that the dataset is better clustered into these K clusters and otherwise it is less likely to be clustered by this K cluster. In this section, the partition coefficient is used to test if i-vectors from different durations follow different distributions. A GMM with two components ($K = 2$) was trained using length normalised i-vectors from short and long utterances. The partition coefficient is then defined as

$$PC = \frac{1}{R}\sum_{i=1}^{N}\sum_{k=1}^{K}\zeta_{ik}^{2} \tag{7-1}$$

where R is the total number of input vecotors and,

$$\zeta_{ik} = \frac{N(\widetilde{\omega}_i|\mu_k', \boldsymbol{\Sigma_k'})}{\sum_{r=1}^{K} N(\widetilde{\omega}_i|\mu_r, \boldsymbol{\Sigma_r})} \tag{7-2}$$

where and $\mu_k^{'}$ and $\boldsymbol{\Sigma_k^{'}}$ are the mean and covariance of each Gaussian mixture component k.

The partition coefficient estimated from the long and short utterances used to generate Figure 7.1 was 0.837 and given that there are two clusters in this case (long and short utterance), the range of values it could have taken is $[0.5,1]$. This relatively high value for the partition coefficient suggests that the i-vectors from long and short duration utterances have high clustering tendency and are likely to have two different distributions.

The comparison of the histograms of the lengths of i-vectors estimated form long and short utterances, and the partition coefficient estimated from the normalised i-vectors corresponding to long and short utterances both suggest that modelling long and short duration i-vectors with the same Gaussian distribution (as is the case in the standard GPLDA model) may be inaccurate. Motivated by these limitations, compensation techniques in the back-end classifier are proposed.

7.2 Compensating duration mismatch in back-ends

This section proposes a Twin Model GPLDA for compensating duration mismatch in short duration ASV. Similar with GPLDA, the proposed Twin-Model GPLDA can also incorporate uncertainty propagation which is beneficial for short duration ASV. The model without uncertainty propagation will be first presented, followed by incorporating uncertainty propagation into this framework.

7.2.1 Proposed Twin Model GPLDA

The assumption here is that i-vectors from the same speaker still share identical normally distributed latent variables, but i-vectors from long and short utterances do not share the same

distribution. Two independent sets of factor analysis hyper-parameters are utilised to account for the mismatch between long and short duration utterances. In other words, instead of one unified 'path', the standard GPLDA model in Figure 7.2(a) is revised as indicated in Figure 7.2(b).

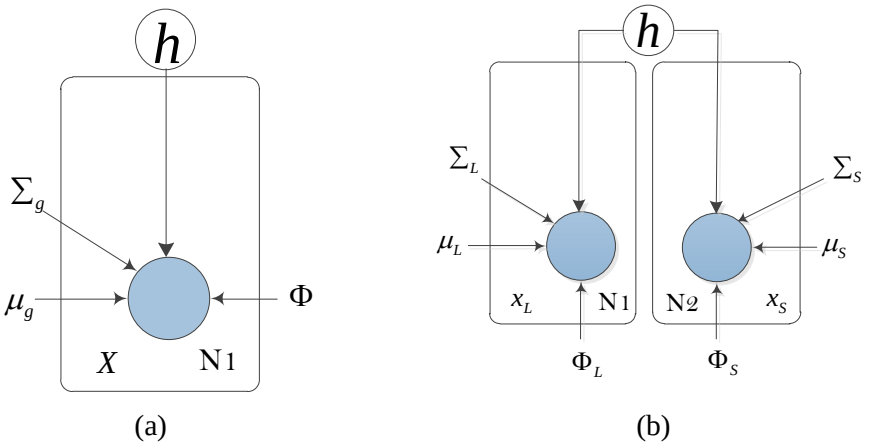


Figure 7.2 Graphical model of (a) standard GPLDA, and (b) Twin Model GPLDA.

The generative equation of the twin model GPLDA is written as:

$$\tilde{\omega} = \begin{cases} \mu_L + \boldsymbol{\Phi_L} h + \varepsilon_L, & \text{for long utterances} \\ \mu_S + \boldsymbol{\Phi_S} h + \varepsilon_S, & \text{for short utterances} \end{cases} \tag{7-3}$$

where $\tilde{\omega}$ denotes the i-vector; μ_L and μ_S are mean vectors for i-vector correspond to long and short utterances, respectively; $\boldsymbol{\Phi_L}$ and $\boldsymbol{\Phi_S}$ are the corresponding factor loading matrices; h is the vector of the normally distributed latent variables shared by all the utterances from the same speaker, and ε_L and ε_S are residuals that are different for different utterances and are assumed to be normally distributed with zero mean and covariances given by the matrices $\boldsymbol{\Sigma_L}$ and $\boldsymbol{\Sigma_S}$ for long and short utterances respectively. Thus, the hyper-parameters, $\theta = \{\mu_L, \boldsymbol{\Phi_L}, \boldsymbol{\Sigma_L}, \mu_S, \boldsymbol{\Phi_S}, \boldsymbol{\Sigma_S}\}$, completely describe the Twin Model GPLDA and will model the differences between long and short durations as well as the within-speaker similarities of i-vectors.

As with the standard GPLDA mentioned in Section 2.3.3, the likelihood-ratio score for speaker verification is obtained by calculating the likelihood of two hypotheses. Specifically, given an enrolment i-vector ω_e from a long utterance, and a test i-vector ω_t from a short utterance, the two hypotheses of interest are: H_s, that ω_e and ω_t share the same latent variable h; and H_d, that ω_e and ω_t are generated by different latent variables. Figure 7.3 shows the graphical models corresponding to the two hypotheses.

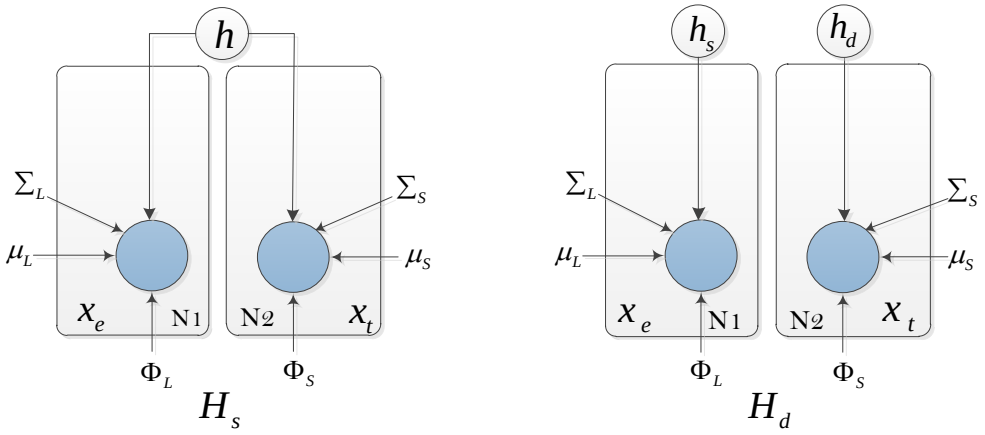


Figure 7.3 Graphical models of the two hypotheses: (a) H_s that the test and enrolment i-vectors are from same speaker (i.e. share latent variables h); and (b) H_d that the test and enrolment i-vectors are from different speakers (i.e. have distinct latent variables h_1 and h_2).

In the Twin Model GPLDA (TM-GPLDA) equations, i-vectors are assumed to be independent conditional on the latent variable h. If two i-vectors share the same latent variables (H_s), factor loading matrices and other parameters can be concatenated to share the same latent variable. Similarly, if two i-vectors are generated by different latent variables (H_d), two latent variables can be stacked to form a vector that has doubled the dimension (see equations (7-4) and (7-5)), as they are assumed to be independent. The factor loading matrices are concatenated into a block diagonal matrix. Thereby we develop the following equations for the two hypotheses:

$$H_s: \omega' = \mu' + \boldsymbol{A}h_s + \varepsilon' \tag{7-4}$$

$$H_d: \omega' = \mu' + \boldsymbol{B}h_d + \varepsilon' \tag{7-5}$$

where,

$$\omega' = \begin{bmatrix}\omega_e \\ \omega_t\end{bmatrix}, \mu' = \begin{bmatrix}\mu_L \\ \mu_S\end{bmatrix}, \ \boldsymbol{A} = \begin{bmatrix}\boldsymbol{\Phi_L} \\ \boldsymbol{\Phi_s}\end{bmatrix}, \boldsymbol{B} = \begin{bmatrix}\boldsymbol{\Phi_L} & 0 \\ 0 & \boldsymbol{\Phi_S}\end{bmatrix}, \varepsilon' = \begin{bmatrix}\varepsilon_L \\ \varepsilon_S\end{bmatrix}.$$

In order to get the likelihood of each hypothesis, we evaluate the following likelihood function:

$$P(\omega'|h_s) = \mathcal{N}\left(\mu' + \boldsymbol{A}h_s, \begin{bmatrix}\boldsymbol{\Sigma_L} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma_S}\end{bmatrix}\right), \tag{7-6}$$

given that

$$P(h_s) = \mathcal{N}(0, \boldsymbol{I}). \tag{7-7}$$

Thus, the marginal likelihood for hypothesis H_s is:

$$P(\omega_e, \omega_t|H_s) = \mathcal{N}\left(\mu', \boldsymbol{A}\boldsymbol{A}^* + \begin{bmatrix}\boldsymbol{\Sigma_L} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma_s}\end{bmatrix}\right). \tag{7-8}$$

Similarly, the marginal likelihood for hypothesis H_d is:

$$P(\omega_e, \omega_t|H_d) = \mathcal{N}\left(\mu', \boldsymbol{B}\boldsymbol{B}^* + \begin{bmatrix}\boldsymbol{\Sigma_L} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma_S}\end{bmatrix}\right). \tag{7-9}$$

The log likelihood ratio is then given by the difference between the logarithms of these two probabilities as:

$$\begin{aligned} Score(x_e, x_t) = log\left(\mathcal{N}\left(\begin{bmatrix}\omega_e \\ \omega_t\end{bmatrix}; \begin{bmatrix}\mu_L \\ \mu_s\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Phi_L}\boldsymbol{\Phi_L^*} + \boldsymbol{\Sigma_L} & \boldsymbol{\Phi_L}\boldsymbol{\Phi_S^*} \\ \boldsymbol{\Phi_S}\boldsymbol{\Phi_S^*} & \boldsymbol{\Phi_s}\boldsymbol{\Phi_L^*} + \boldsymbol{\Sigma_s}\end{bmatrix}\right)\right) \\ - log\left(\mathcal{N}\left(\begin{bmatrix}\omega_e \\ \omega_t\end{bmatrix}; \begin{bmatrix}\mu_L \\ \mu_s\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Phi_L}\boldsymbol{\Phi_L^*} + \boldsymbol{\Sigma_L} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi_s}\boldsymbol{\Phi_S^*} + \boldsymbol{\Sigma_s}\end{bmatrix}\right)\right). \end{aligned} \tag{7-10}$$

It can be seen that the scoring equation given by (7-10) has a similar structure to that of equation (2-26). Also, note that if we set $\boldsymbol{\Phi_L} = \boldsymbol{\Phi_s}$, $\mu_L = \mu_s$ and $\boldsymbol{\Sigma_L} = \boldsymbol{\Sigma_s}$ in (7-10), it returns to the standard GPLDA scoring found in equation (2-12).

7.2.2 Twin Model GPLDA Parameter Estimation

The hyper-parameters of standard GPLDA are estimated using the EM algorithm (described in Appendix A) from background i-vectors. In twin model GPLDA, two sets of hyper-parameters associated with both long and short i-vectors from the same speaker should be tied to one unique set of speaker latent variables. This is different to standard GPLDA parameter estimation. The derivation of the EM algorithm for this particular GPLDA will be shown below.

Let $\theta = \{\mu_L, \boldsymbol{\Phi_L}, \boldsymbol{\Sigma_L}, \mu_S, \boldsymbol{\Phi_S}, \boldsymbol{\Sigma_S}\}$ denote the parameters that need to be estimated. Let x_L and x_S represent i-vectors from long and short utterances, respectively, and let h represent the latent variable in equation (7-3). In the standard GPLDA, i-vectors from one speaker will form a class and share one latent variable. The posterior expectation $E[h]$ is then obtained by using the factor analysis model as,

$$E[h] = \left(\boldsymbol{\Phi}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi} + \boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}^*\boldsymbol{\Sigma}^{-1}\left(\widetilde{\omega} - \mu_g\right). \tag{7-11}$$

In the proposed twin model GPLDA, there are both long and short duration i-vectors from the same speaker that share the same latent variables. To estimate the model parameters, merged i-vectors will be created by concatenating one i-vector from a long utterance with one from a short utterance from the same speaker and denoted by $\widetilde{\omega}_m = \begin{bmatrix}\widetilde{\omega}_e \\ \widetilde{\omega}_t\end{bmatrix}$. The E-step is then formulated as:

$$E[h] = \left(\boldsymbol{A}^*\boldsymbol{\Sigma'}^{-1}\boldsymbol{A} + \boldsymbol{I}\right)^{-1}\boldsymbol{A}^*\boldsymbol{\Sigma'}^{-1}(\widetilde{\omega}_m - \mu') \tag{7-12}$$

$$E[hh^T] = \left(\boldsymbol{A}^*\boldsymbol{\Sigma'^{-1}}\boldsymbol{A} + \boldsymbol{I}\right)^{-1} + E[h]E[h^*] \tag{7-13}$$

where, $\boldsymbol{\Sigma'} = \begin{bmatrix} \boldsymbol{\Sigma_L} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma_S} \end{bmatrix}$, and $\widetilde{\omega}_m = \begin{bmatrix} \widetilde{\omega}_L \\ \widetilde{\omega}_S \end{bmatrix}$.

In the M-step, the auxiliary function to be optimised is

$$Q(\theta, \theta_{old}) = \sum_i \sum_j \int p(h_i|\chi, \theta_{old}) log\left[p\left(\widetilde{\omega}_{ij}\middle|h_i\right)p(h_i)\right]dh_i \tag{7-14}$$

where h_i denotes the latent variables corresponding to the i^{th} speaker, $\widetilde{\omega}_{ij}$ denotes the j^{th} i-vectors from the i^{th} speaker, χ denotes i-vectors from all training speakers, and θ_{old} denotes the model hyper-parameters from the previous iteration of the EM algorithm.

By taking the derivatives with respect to θ and setting them to zero, we obtain the following update equations:

$$\mu_L = \frac{1}{N_L}\sum_{i,j} \widetilde{\omega}_{L_{ij}} \ , \tag{7-15}$$

$$\boldsymbol{\Phi_L} = \left(\sum_{i,j}\left(\widetilde{\omega}_{L_{ij}} - \mu_L\right)E[h_i^*]\right)\left(\sum_{i,j} E[h_i {h_i}^*]\right)^{-1} , \tag{7-16}$$

$$\boldsymbol{\Sigma_L} = \frac{1}{N_L}\sum_{i,j}\left[\left(\widetilde{\omega}_{L_{ij}} - \mu_L\right)\left(\widetilde{\omega}_{L_{ij}} - \mu_L\right)^* - \boldsymbol{\Phi_L}E[h_i]\left(\widetilde{\omega}_{L_{ij}} - \mu_L\right)\right] \tag{7-17}$$

where $\widetilde{\omega}_{L_{ij}}$ denotes the i-vector corresponding to the j^{th} long utterance from the i^{th} speaker, and

$$\mu_S = \frac{1}{N_S}\sum_{i,j} \widetilde{\omega}_{S_{ij}} , \tag{7-18}$$

$$\boldsymbol{\Phi_s} = \left(\sum_{i,j} \left(\widetilde{\omega}_{s_{ij}} - \mu_s \right) E[h_i^*] \right) \left(\sum_{i,j} E[h_i {h_i}^*] \right)^{-1}, \tag{7-19}$$

$$\boldsymbol{\Sigma_s} = \frac{1}{N_s} \sum_{i,j} \left[\left(\widetilde{\omega}_{s_{ij}} - \mu_s \right) \left(\widetilde{\omega}_{s_{ij}} - \mu_s \right)^* - \boldsymbol{\Phi_s} E[h_i] \left(\widetilde{\omega}_{s_{ij}} - \mu_s \right) \right]. \tag{7-20}$$

where $\widetilde{\omega}_{s_{ij}}$ denotes the i-vector corresponding to the j^{th} short utterance from the i^{th} speaker.

7.2.3 Incorporating uncertainty propagation

7.2.3.1 Uncertainty propagation in GPLDA

As illustrated in Section 6.1 by equations (6-1) to (6-4), a vector representation like the i-vector is the mean of posterior probabilities of the latent variables. Information of a given utterance is propagated through its zeroth- and first-order statistics. Specifically, once the model is trained, the posterior covariance of the latent variables is given as equation (6-1), in which the posterior covariance is determined by zeroth-order statistics N. As mentioned in Section 6.1, for short utterances, the zeroth-order statistics are smaller, which makes the vector representations, i.e. the i-vector, less representative of corresponding utterance than that of long duration utterance. The uncertainty of the distribution is measured by its covariance.

In [88], this uncertainty is propagated into the GPLDA model by adding a factor corresponding to the uncertainty covariance in a factor analysis model. The GPLDA model is the same as that found in Section 2.3. The generative model behind the GPLDA with uncertainty propagation [88] is

$$\widetilde{\omega}_{ij} = \mu_g + \boldsymbol{\Phi} h_i + \boldsymbol{U_{ij}} o_{ij} + \varepsilon_{ij} \tag{7-21}$$

where $\boldsymbol{U_j U_j^*}$ is the Choleskey decomposition of the posterior covariance matrix associated with corresponding the i-vector, and o_j is a hidden variable having a standard normal distribution.

The other parameters are the same as those used in equation (2-11), where μ_g is the mean of pre-processed i-vector, $\boldsymbol{\Phi}$ is a factor loading matrix, h_i is a vector of latent variables with a standard Gaussian distribution of $N(0, \boldsymbol{I})$, and ε_{ij} is a residual term that is assumed to be Gaussian with zero mean and a full covariance matrix denoted by Σ_g. In the phase of training the parameters, latent variables h_i and x_j are stacked together to form one latent variable and corresponding factor loadings are also stacked [88, 115] and EM algorithm in Appendix A is used to train those parameters.

7.2.3.2 Uncertainty propagation in twin model GPLDA

Similar to the uncertainty propagation in GPLDA mentioned in previous section, the generative equation of TM-GPLDA is modified from equation (7-3) as:

$$\omega_{ij} = \begin{cases} \mu_L + \boldsymbol{\Phi_L} h_i + \boldsymbol{U_{ij,L}} o_{ij,L} + \varepsilon_{ij,L}, \text{for long utterances} \\ \mu_S + \boldsymbol{\Phi_S} h_i + \boldsymbol{U_{ij,S}} o_{ij,S} + \varepsilon_{ij,S}, \text{for short utterances} \end{cases} \tag{7-22}$$

where L denotes the long utterance and S the short utterance classes; μ_L, μ_S, $\boldsymbol{\Phi_L}$, $\boldsymbol{\Phi_S}$ and h_i are as defined for the standard TM-GPLDA given in (7-3); the residual terms $\varepsilon_{ij,L}$ and $\varepsilon_{ij,S}$ follow $\mathcal{N}(0, \boldsymbol{\Sigma_L})$ and $\mathcal{N}(0, \boldsymbol{\Sigma_S})$ respectively; $o_{ij,L}$ and $o_{ij,S}$ are latent variables having standard normal distributions; $\boldsymbol{U_{ij,L}}\boldsymbol{U^*_{ij,L}}$ and $\boldsymbol{U_{ij,S}}\boldsymbol{U^*_{ij,S}}$ are the Choleskey decompositions of the posterior covariance matrices associated with the corresponding i-vectors. The EM algorithm is used to estimate the model hyper-parameters and the formulations are developed in following sections.

7.2.3.3 Expectation step

In the E-step, the posterior probabilities of latent variables are estimated. According to Bayes rule, the posterior distribution is proportional with the production of likelihood and prior, which is expressed as

$$p(h_i|\omega_i,\theta) \propto p(\omega_i|h_i,\theta)p(h_i) \tag{7-23}$$

where ω_i denotes the set of i-vectors from the i^{th}speaker. The likelihood term is expanded as:

$$\begin{aligned} p(\omega_i|h_i,\theta) &= p\big(\omega_i|h_i, o_{i1,L},\dots,o_{iJ_i,L}, o_{i1,S},\dots,o_{iJ_i,S},\theta\big) \\ &= \prod_i \prod_j \prod_{k\in L,S} \int p\big(\omega_{ij,k}|h_i,o_{ij,k},\theta\big)p\big(o_{ij,k}\big)do_{ij,k} \end{aligned} \tag{7-24}$$

where

$$p\big(\omega_{ij,k}|h_i,o_{ij,k},\theta\big) = \mathcal{N}\big(\omega_{ij,k}|\mu_k + \boldsymbol{\Phi_k}h_i + \boldsymbol{U_{ij,k}}o_{ij,k},\Sigma_k\big). \tag{7-25}$$

and the prior distribution of latent variable *o* is a standard normal distribution. According to the convolution of Gaussian kernel [142], the integral in (7-24) is evaluated as $\mathcal{N}(\omega_{ij,k}|\mu_k + \boldsymbol{\Phi_k}h_i, \boldsymbol{\Sigma_k} + \boldsymbol{U_{ij,k}}\boldsymbol{U^*_{ij,k}})$. Notice that the term $\boldsymbol{U_{ij,k}}\boldsymbol{U^*_{ij,k}}$ is now part of the residual covariance.

As in [63], we write the likelihood term as a stacked equation as follows:

$$\begin{bmatrix} \widetilde{\omega}_{i1,L} \\ \dots \\ \widetilde{\omega}_{iJ_{iL},L} \\ \widetilde{\omega}_{i1,S} \\ \dots \\ \widetilde{\omega}_{iJ_{iS},S} \end{bmatrix} = \begin{bmatrix} \mu_L \\ \dots \\ \mu_L \\ \mu_S \\ \dots \\ \mu_S \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi_L} \\ \dots \\ \boldsymbol{\Phi_L} \\ \boldsymbol{\Phi_S} \\ \dots \\ \boldsymbol{\Phi_S} \end{bmatrix} h_i + \begin{bmatrix} \tilde{\varepsilon}_{i1,L} \\ \dots \\ \tilde{\varepsilon}_{iJ_{iL},L} \\ \tilde{\varepsilon}_{i1,S} \\ \dots \\ \tilde{\varepsilon}_{iJ_{iS},S} \end{bmatrix} \tag{7-26}$$

where subscript J_{iL} means the number of long duration utterances of i^{th} speaker and subscript J_{iS} means the number of short duration utterances of i^{th} speaker. This can then be written as a Gaussian kernel $\mathcal{N}(\omega_i|\mu_i + \boldsymbol{\widetilde{\Phi}_i}h_i, \boldsymbol{\widetilde{\Sigma}_i})$, where

$$\omega_i = \begin{bmatrix} \widetilde{\omega}_{i1,L} \\ \dots \\ \widetilde{\omega}_{iJ_{iL},L} \\ \widetilde{\omega}_{i1,S} \\ \dots \\ \widetilde{\omega}_{iJ_{iS},S} \end{bmatrix}, \mu_i = \begin{bmatrix} \mu_L \\ \dots \\ \mu_L \\ \mu_S \\ \dots \\ \mu_S \end{bmatrix}, \widetilde{\Phi}_i = \begin{bmatrix} \boldsymbol{\Phi_L} \\ \dots \\ \boldsymbol{\Phi_L} \\ \boldsymbol{\Phi_S} \\ \dots \\ \boldsymbol{\Phi_S} \end{bmatrix}, \tag{7-27}$$

$$\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{i}} = \begin{bmatrix} \boldsymbol{\Sigma_L} + \boldsymbol{U_{ij,L}}\boldsymbol{U_{ij,L}^*} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \boldsymbol{\Sigma_S} + \boldsymbol{U_{ij,S}}\boldsymbol{U_{ij_{iS},S}^*} \end{bmatrix}. \tag{7-28}$$

The posterior probability is then proportional to a product of two Gaussian kernels, expressed as:

$$p(h_i|\omega_i, \theta) \propto \mathcal{N}\left(\omega_i|\mu_i + \widetilde{\boldsymbol{\Phi}}_{\boldsymbol{i}} h_i, \tilde{\Sigma}_i\right)\mathcal{N}(h_i|0, \boldsymbol{I}). \tag{7-29}$$

Based on the fact that the outcome of product of two Gaussians is still a Gaussian, the first and second moment of the posterior probability (7-29) are expressed as

$$E(h) = \left(\boldsymbol{I} + \widetilde{\boldsymbol{\Phi}}_{\boldsymbol{i}}{}^{*}\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{i}}^{-\mathbf{1}}\widetilde{\boldsymbol{\Phi}}_{\boldsymbol{i}}\right)^{-1}\widetilde{\boldsymbol{\Phi}}_{\boldsymbol{i}}{}^{*}\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{i}}^{-\mathbf{1}}(\omega_i - \mu_i), \tag{7-30}$$

$$E(h_i h_i^*) = \left(\boldsymbol{I} + \widetilde{\boldsymbol{\Phi}}_{\boldsymbol{i}}{}^{*}\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{i}}^{-\mathbf{1}}\widetilde{\boldsymbol{\Phi}}_{\boldsymbol{i}}\right)^{-1} + E(h_i)E(h_i^*). \tag{7-31}$$

Similarly, the first and second moment of the posterior probability of latent variable $o_{ij,k}$ in (7-22) are calculated as:

$$E\left(o_{ij,k}\right) = \left[\boldsymbol{I} + \boldsymbol{U_{ij,k}^*}(\boldsymbol{\Phi_k}\boldsymbol{\Phi_k^*} + \boldsymbol{\Sigma_k})^{-1}\boldsymbol{U_{ij,k}}\right]^{-1}\boldsymbol{U_{ij,k}^*}(\boldsymbol{\Phi_k}\boldsymbol{\Phi_k^*} + \boldsymbol{\Sigma_k})^{-1}\left(\widetilde{\omega}_{ij,k} - \mu_k\right), \tag{7-32}$$

$$E\left(o_{ij,k}o_{ij,k}^*\right) = \left[\boldsymbol{I} + \boldsymbol{U_{ij,k}^*}(\boldsymbol{\Phi_k}\boldsymbol{\Phi_k^*} + \boldsymbol{\Sigma_k})^{-1}\boldsymbol{U_{ij,k}}\right]^{-1} + E\left(o_{ij,k}\right)E\left(o_{ij,k}^*\right) \tag{7-33}$$

where $k \in \{L, S\}$.

7.2.3.4 Maximization step

The auxiliary function of EM algorithm is of the form:

$$Q(\theta, \theta_{old}) = \int p(H|\omega, \theta_{old}) log[p(\omega|H)p(H)]dH \tag{7-34}$$

where H denotes the collected latent variables including h and o, which they are independent of each other, ω denotes i-vectors, and θ_{old} denotes the model hyper-parameters from the previous

iteration of the EM algorithm. This is regarded as a lower bound of log-likelihood of the observable data and each step will increase the log-likelihood, leading to a local optimum of parameters. By observing the generative model, it can be found that given the latent variable, the observable variables are independent, and the parameters are associated with the corresponding long or short class similar with TM-GPLDA in Section 7.2.2. This indicates that the auxiliary function is a linear combination of different classes. In the M-step, we optimise the auxiliary function:

$$\tilde{Q}(\theta, \theta_{old}) = \sum\nolimits_k Q(\theta_k, \theta_{old}) \tag{7-35}$$

where

$$Q(\theta_k, \theta_{old}) = \int p(H|\omega, \theta_{old}) log[p(\omega_k|H)p(H)]\, dH. \tag{7-36}$$

where ω_k denotes i-vectors from class k (either long or short utterance, i.e. $k \in \{L, S\}$ (what is L and what is S)). The auxiliary function is written as a summation of the long and short classes in terms of their parameters, which means that these two sets of parameters can be updated separately. By setting the derivative with respect to each parameter to zero, and after some algebraic manipulations, the update equations can be written as:

$$\boldsymbol{\Phi_k} = \left\{ \sum_{ij} \left[\omega_{ij,k} - \mu_k - \boldsymbol{U_{ij,k}} E\left(o_{ij,k}\right) \right] E(h_i) \right\} \left[\sum_{ij} E(h_i h_i^*) \right]^{-1}, \tag{7-37}$$

$$\begin{aligned} \boldsymbol{\Sigma_k} = \frac{1}{\sum_i J_{i,k}} \Bigg\{ \sum\nolimits_{ij} \Big[\left(\omega_{ij,k} - \mu_k\right) \left(\omega_{ijk} - \mu_k\right)^* \\ - \left[\boldsymbol{\Phi_k} E(h_i) + \boldsymbol{U_{ij,k}} E\left(o_{ij,k}\right) \right] \left(\omega_{ij,k} - \mu_k\right) \Big] \Bigg\}. \end{aligned} \tag{7-38}$$

7.2.3.5 Scoring

Based on TM-GPLDA with an uncertainty propagation model, given an enrolment i-vector ω_e and a test i-vector ω_t from a trial, the log-likelihood ratio between the hypothesis that the two i-vectors are from the same speaker (H_s) versus the hypothesis that they are from different speakers (H_d) is calculated as:

$$Score(\omega_e, \omega_t) = log \frac{p(\omega_e, \omega_t|H_s)}{p(\omega_e, \omega_t|H_d)} \tag{7-39}$$

where

$$p(\omega_e, \omega_t|H_s) = \int p(\omega_e, \omega_t|h, o_e, o_t)p(h)p(o_e)p(o_t)dhdo_e do_t, \tag{7-40}$$

$$\begin{aligned} &p(\omega_e, \omega_t|H_d) \\ &= \int p(\omega_e|h_e, o_e)p(h_e)p(o_e)\, dh_e do_e \int p(\omega_t|h_t, o_t)\, p(h_t)p(o_t)dh_t do_t. \end{aligned} \tag{7-41}$$

Figure 7.4 shows the graphical models corresponding to the two hypotheses. The evaluation procedure is similar to the E-step. The likelihood terms for each hypothesis are calculated as:

$$p(\omega_e, \omega_t|H_s) = \mathcal{N}\left(\begin{bmatrix}\omega_e \\ \omega_t\end{bmatrix} | \begin{bmatrix}\mu_L \\ \mu_S\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma_L} + \boldsymbol{\Phi_L}\boldsymbol{\Phi_L^*} + \boldsymbol{U_e}\boldsymbol{U_e^*} & \boldsymbol{\Phi_L}\boldsymbol{\Phi_S^*} \\ \boldsymbol{\Phi_S}\boldsymbol{\Phi_L^*} & \boldsymbol{\Sigma_S} + \boldsymbol{\Phi_S}\boldsymbol{\Phi_S^*} + \boldsymbol{U_t}\boldsymbol{U_t^*}\end{bmatrix}\right), \tag{7-42}$$

$$p(\omega_e, \omega_t|H_d) = \mathcal{N}\left(\begin{bmatrix}\omega_e \\ \omega_t\end{bmatrix} | \begin{bmatrix}\mu_L \\ \mu_S\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma_L} + \boldsymbol{\Phi_L}\boldsymbol{\Phi_L^*} + \boldsymbol{U_e}\boldsymbol{U_e^*} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma_S} + \boldsymbol{\Phi_S}\boldsymbol{\Phi_S^*} + \boldsymbol{U_t}\boldsymbol{U_t^*}\end{bmatrix}\right). \tag{7-43}$$

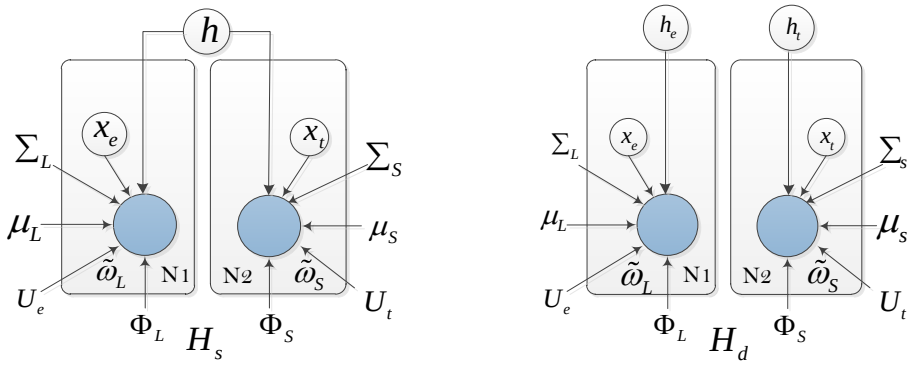


Figure 7.4 Graphical models showing the hypotheses (a) the test and enrolment i-vectors are from same speaker, i.e., share latent variables y; and (b) that the test and enrolment i-vectors are from different speakers, i.e., have distinct latent variables, h_e and h_t.

7.2.3.6 Scaling covariance

From Section 7.3.5, it can be seen that the uncertainty of an i-vector is propagated into the covariance of the marginal distribution in the calculation of the posterior probability and final score. By marginalising latent variables in (7-24), the posterior probability is evaluated as $\mathcal{N}(\tilde{\omega}_{ij,k}|\mu_k, \boldsymbol{\Sigma_k} + \boldsymbol{\Phi_k}\boldsymbol{\Phi_k^*} + \boldsymbol{U_{ij,k}}\boldsymbol{U_{ij,k}^*})$. This means the covariance of posterior probability consists of three terms, which are $\boldsymbol{\Sigma_k}$, $\boldsymbol{\Phi_k}\boldsymbol{\Phi_k^*}$ and the uncertainty of corresponding i-vector $\boldsymbol{U_{ij,k}}\boldsymbol{U_{ij,k}^*}$. As the uncertainty tends to be larger with decreasing duration, element value of the covariance of the latent variable in the total variability model of short utterances become much larger than those of $\boldsymbol{\Sigma_k}$, $\boldsymbol{\Phi_k}\boldsymbol{\Phi_k^*}$ in the hyper-parameters training phase, making the model ignore intra- and inter-speaker variability which is supposed to be captured by $\boldsymbol{\Phi_k}\boldsymbol{\Phi_k^*}$ and $\boldsymbol{\Sigma_k}$ and leading to poor modelling ability of the GPLDA. To solve this problem, scaling factors are introduced into TM-GPLDA with uncertainty propagation. The generative model is then revised from equation (7-22) to become:

$$\tilde{\omega}_{ij} = \begin{cases} \mu_L + \boldsymbol{\Phi_L} h_i + \lambda_L \boldsymbol{U_{ij,L}} o_{ij,L} + \varepsilon_{ij,L}\,, & \text{for long utterances} \\ \mu_S + \boldsymbol{\Phi_S} h_i + \lambda_S \boldsymbol{U_{ij,S}} o_{ij,S} + \varepsilon_{ij,S}, & \text{for short utterances} \end{cases} \tag{7-44}$$

where λ_L and λ_S are scaling factors, the value of which are empirically determined. Our basic assumption is that the structure of the covariance matrix for a given class (long or short) is important, and that scaling factors are introduced to balance the contributions from different uncertainties (e.g., the uncertainty from a single i-vector or from the residual in equation (7-44)). The validity of this assumption is supported by [143], where an identical covariance can be shared by utterances that are of roughly the same length. Instead of using the same covariance, we relax this condition to instead have the same scaling factor. The scaling factors can be directly absorbed into the $\boldsymbol{U_{ij,k}}$ term from equation (7-22) and the same modelling, training, scoring equations can be used.

7.3 Experiments

7.3.1 Experiments with Twin Model GPLDA without uncertainty propagation

A number of experiments were carried out to analyse the effectiveness of the proposed twin model GPLDA. The 8CONV-10SEC condition (CC5) and additional 8CONV-5SEC and 8CONV-3SEC of the NIST SRE' 2010 [13] was chosen for these experiments.

The baseline system is an i-vector/GPLDA system, the same as in Sections 3.3 and 4.2. The parameters of the GPLDA model were trained on utterances with duration varying from 3 seconds to 2.5 minutes. Short duration utterances are truncated from original long duration utterances, which are approximately 2.5 minutes long. For the proposed TM-GPLDA, utterances in long duration class are typically 2.5 minutes, and duration in short duration class varies from 3 seconds to 20 seconds. For all results presented in the table, speakers were enrolled using 8 utterances of about 2.5 minutes, while the test segments were set to 10 seconds, 5 seconds and 3 seconds.

Table 7.1-2 summarise the results of the standard GPLDA system trained on utterances of varying durations in terms of EER and Figure 7.5-6 present the results in terms of MinDCF. The

results are consistent with those in [121], which suggest that for short duration speaker verification, it is not optimal to use full utterances in the GPLDA hyper-parameters training phase. It is observed that using short development utterances (e.g., 15 seconds or shorter) benefits the extremely shorter test scenarios of 5 seconds and 3 seconds. This may be because speaker factors estimated from long utterances do not adequately characterise the short duration utterances. By making a compromise between long and short durations by using relatively short (15 seconds) development utterances, a better model that characterises both long and short utterances might be obtained. These results reinforce the clear need to have a model that can take into account differences between short and long utterances.

Table 7.1 also shows the accuracies of speaker verification systems employing the twin model GPLDA. As there are two sets of hyper-parameters in the twin model GPLDA, i-vectors from both long and short (truncated) utterances are needed to train the parameters. Thus, in this experiment, i-vectors from full 2.5 minutes utterances are used along with i-vectors from truncated utterances of varying durations (given in Table 7.1 and 7.2) to estimate the two sets of hyper-parameters. Enrolment and test utterances are identical to the ones used with the standard GPLDA. It is clear from the results that the proposed method outperforms the baseline approach for all three short test utterance durations. For the 5 second and 3 second test conditions, the best performance was obtained when using truncated utterances of 10 to 20 seconds duration to train the twin model GPLDA hyper-parameters. Relative improvements of 12.4% and 21.2% were observed for the 5 second and 3 second conditions respectively when comparing the proposed twin model GPLDA to the baseline. When compared with the best results obtained by standard GPLDA, 12.4% and 14.7% relative improvements are observed for the 5 second and 3 second conditions respectively. In Figure 7.5 and 7.6, MinDCF values of TM-GPLDA systems are lower than those of GPLDA, which also confirms that TM-GPLDA has better performance.

Table 7.1 Performance (EER %) using the standard and the proposed twin model GPLDA systems on the NIST SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions (male speakers)

Training data	Test duration					
	Standard GPLDA			Twin Model GPLDA		
	8CONV-10SEC	8CONV-5SEC	8CONV-3SEC	8CONV-10SEC	8CONV-5SEC	8CONV-3SEC
3SEC	14.41	18.14	22.47	8.08	11.45	**13.93**
5SEC	12.45	16.47	20.91	6.94	11.01	14.63
10SEC	8.48	14.26	18.87	5.74	10	14.15
15SEC	7.77	12.85	18.27	5.21	9.73	14.12
20SEC	7.99	12.29	16.33	**4.78**	**9.4**	14.09
2.5min	5.03	10.73	17.68	5.03	10.73	17.68

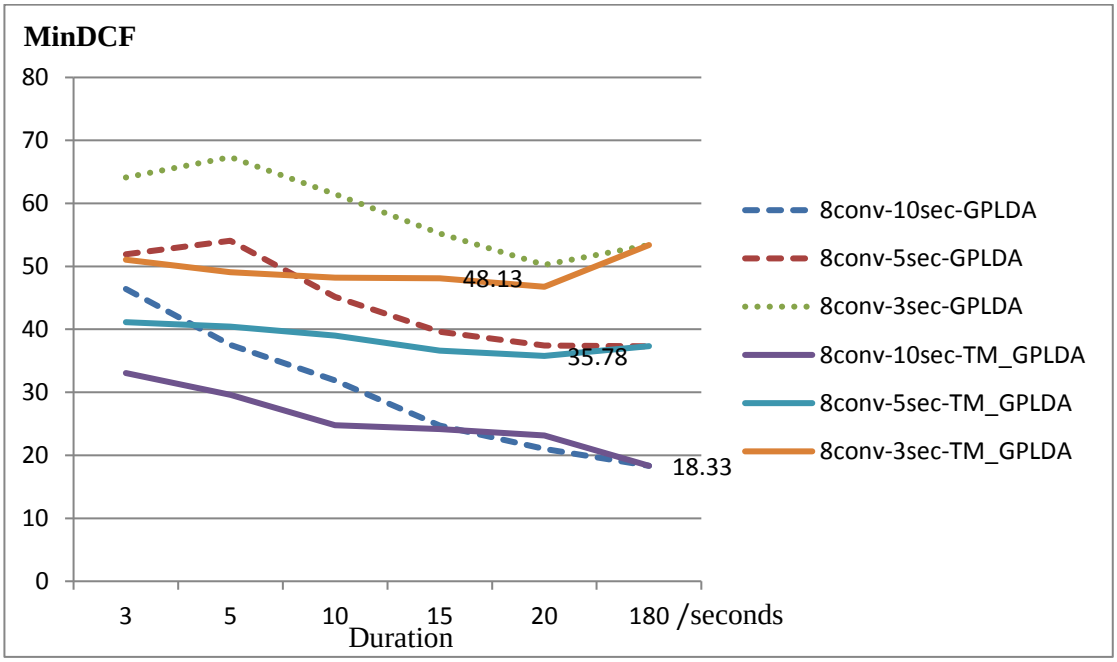


Figure 7.5 Performance (MinDCF %) using standard and the Twin-Model GPLDA systems on the NIST SRE'10 8CONV-10SEC, and additional 5SEC and 3SEC conditions (male part).

It was observed that when the duration of training utterances falls below 10 seconds, the performance of the overall system drops again for 5 and 10 seconds test conditions. This is probably the result of not having sufficient training data frames for the estimation of the model parameters. Similarly, although the improvement in the 10 second condition was minor compared to the baseline, the trends remain consistent. One reason why the improvement is so small may be because the proposed method is more useful when dealing with the more severe cases such as the 5 second and 3 second tests than with the slightly longer duration 10 second tests. For longer utterances, the two sets of hyper-parameters of the discriminative GPLDA will be similar and the proposed method will not be much more efficient than the standard GPLDA because the mismatch between enrolment and test utterances is not as severe.

Table 7.2 Performance (EER %) using the standard and the proposed twin model GPLDA systems on the NIST SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions (female speakers)

Training data	Test duration					
	Standard GPLDA			Twin Model GPLDA		
	8CONV-10SEC	8CONV-5SEC	8CONV-3SEC	8CONV-10SEC	8CONV-5SEC	8CONV-3SEC
3SEC	14.86	20.25	22.2	10.42	15.27	18.54
5SEC	11.57	17.91	25.99	9.55	13.78	18.02
10SEC	9.97	15.72	21.27	7.59	13.29	17.6
15SEC	8.83	14.08	20.43	7.68	12.27	**15.72**
20SEC	7.23	13.27	17.37	**6.86**	**11.95**	16.57
2.5min	6.16	12.43	18.90	6.16	12.43	18.90

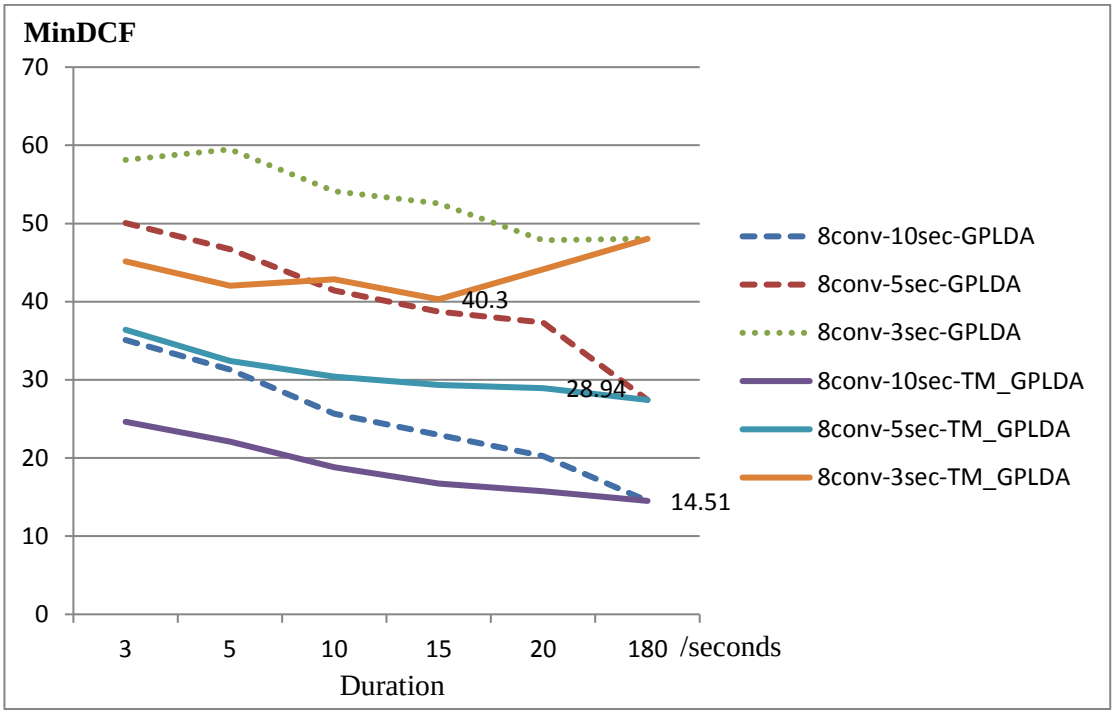


Figure 7.6 Performance (MinDCF %) using standard and the Twin-Model GPLDA systems on the NIST SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions (female part)

7.3.2 Experiments for uncertainty propagation

Following the previous section, this section is to test the effects of uncertainty propagation in TM-GPLDA. By using the uncertainty propagation, i-vectors were radially Gaussianised followed by length normalisation [64]. The post-processing procedure for covariances were the same as in [88, 140], whereby the covariances were transformed by LDA, whitened, and then scaled by the norm of the corresponding i-vector. To be consistent with [116] and reduce the number of covariances stored in memory, the TM-GPLDA model with uncertainty propagation was trained using long utterances as well utterances obtained by truncating the long utterances into 20 second segments. To reduce the total number of short duration utterances, around 25% of the total short duration utterances extracted from each long utterance were randomly selected.

Table 7.3 presents the performances when using different scaling factors. To simplify the experiments, we used equal scaling factors for both long and short utterances and a limited number of values were compared. From the table we can find that results with scaling factors gain significant improvements for all three conditions compared to those with no scaling. It is also found that when increasing the scaling factors beyond 1/6, the model tends to become overfitted and the results become worse. As such, the scaling factors are then selected as $1/2$ for the rest of the experiments.

Table 7.4 and Table 7.5 summarise performances with difference number of training speakers in the training data in EER and MinDCF. The best results when using different numbers of training speakers are shown in bold typeface. It was found that TM-GPLDA works better in severely mismatched conditions like the 5 seconds and 3 seconds conditions. For example, the best results for 3 seconds condition for male and female are from TM-GPLDA_UP system when there are only 500 speakers in training data. It is also true for female speech data when there are 1000 speakers in training data. It is also found that when there are enough speakers (e.g., 2000), TM-GPLDA with uncertainty propagation outperformed the baseline, the standard TM-GPLDA and GPLDA with uncertainty propagation. But the improvements are not significant. In some cases, results with uncertainty propagation in TM-GPLDA are inferior compared with other systems in tables. The reason for this may be that the uncertainty associated with i-vectors estimated from short duration utterances is much more dispersed than that from long duration utterances, or that insufficient short utterances were used in training the model as we only use 25% of total short duration utterances in these experiments due to the memory constraints during system implementation. To guarantee superior results, larger number of utterances is needed, and this can be supported by the facts that results of TM-GPLDA in Table 7.3 and 7.4 are inferior when compared with those in Table 7.1 and 7.2 in same conditions. Larger number of short duration utterances should be tested and is planned as future experimental work.

Table 7.3 Performance (EER %) of TM-GPLDA with uncertainty propagation on SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions (female speakers only) with different values of scaling factors (λ_L, λ_S) with 500 speakers in training data

(λ_L, λ_S)	Test duration		
	10s	5s	3s
$(1,1)$	8.04	13.20	17.31
$(\mathbf{1/2}, \mathbf{1/2})$	6.60	11.75	16.10
$(1/3, 1/3)$	6.60	11.48	17.09
$(1/6, 1/6)$	6.36	12.06	17.59
$(1/12, 1/12)$	6.33	12.06	17.27

Table 7.4 Performance (EER %) of GPLDA, GPLDA with uncertainty propagation (GPLDA_UP), TM-GPLDA and TM-GPLDA with uncertainty propagation (TM-GPLDA_UP) on SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions, with different number

	EER%					
	Female			Male		
	10s	5s	3s	10s	5s	3s
500 speakers						
G-PLDA	6.00	13.58	21.74	6.35	13.39	19.86
G-PLDA_UP	**5.89**	**12.80**	18.57	**5.92**	**11.86**	17.47
TM-GPLDA	7.07	14.53	18.68	7.25	12.31	18.69
TM-GPLDA_UP	7.65	13.39	**18.39**	7.28	11.92	**16.55**
1000 speakers						
G-PLDA	5.42	12.94	20.55	5.32	10.93	17.58
GPLDA_UP	**4.97**	**11.20**	**17.11**	5.14	11.49	16.93
TM-GPLDA	6.97	13.38	17.33	6.27	10.76	16.91
TM-GPLDA_UP	7.55	13.58	18.13	5.82	**10.81**	**16.47**
2000 speakers						
G-PLDA	5.54	12.64	20.14	5.24	10.74	16.85
GPLDA_UP	**4.83**	11.61	17.44	**5.14**	10.69	16.48
TM-GPLDA	5.89	11.83	16.52	5.56	**10.54**	15.23
TM-GPLDA_UP	6.08	**11.58**	**16.06**	6.56	10.89	**15.52**

Table 7.5 Performance (MinDCF%) of GPLDA, GPLDA with uncertainty propagation (GPLDA_UP), TM-GPLDA and TM-GPLDA with uncertainty propagation (TM-GPLDA_UP) on SRE'10 8CONV-10SEC and additional 5SEC and 3SEC conditions, with different number

	MinDCF%					
	Female			Male		
	10s	5s	3s	10s	5s	3s
500 speakers						
G-PLDA	17.81	40.05	61.51	19.43	40.77	57.70
G-PLDA_UP	**16.97**	**36.13**	56.88	**18.29**	**34.11**	51.82
TM-GPLDA	23.06	42.91	57.73	21.49	39.74	51.62
TM-GPLDA_UP	23.07	40.55	**55.58**	23.33	37.62	52.01
1000 speakers						
G-PLDA	16.70	38.99	56.04	15.73	34.04	50.06
GPLDA_UP	**15.29**	**34.40**	51.29	**15.80**	**31.75**	49.12
TM-GPLDA	20.48	35.29	**49.88**	18.43	32.73	**44.87**
TM-GPLDA_UP	23.28	38.11	51.93	19.25	35.65	48.23
2000 speakers						
G-PLDA	16.71	35.93	54.73	**13.63**	32.20	45.52
GPLDA_UP	**14.28**	**33.65**	49.83	15.80	32.37	45.58
TM-GPLDA	18.80	34.70	**47.22**	16.95	**30.38**	**45.25**
TM-GPLDA_UP	18.95	34.81	47.89	18.20	34.67	46.17

7.4 Summary

In this chapter, it is first shown that duration mismatch propagates into classifiers and then the Twin Model Gaussian Probabilistic Linear Discriminative Analysis (TM-GPLDA) is proposed to deal with this duration mismatch in the case of long duration enrolment and short duration test utterances speaker verification. Experimental results demonstrate that vector representations (i.e. i-vector) of long and short utterances have distinct distributions which contradict the assumptions in the standard GPLDA model. Hence, the standard GPLDA model is modified to have two separated generative paths and jointly trained by i-vectors from long utterances and

short utterances. The efficacy of the proposed technique is validated on the NIST SRE'10 8CONV-10SEC condition and additional shorter duration conditions using the truncated 5 and 3 seconds test data. Furthermore, similar to the uncertainty propagation in GPLDA, uncertainty propagation into TM-GPLDA is also developed to take into account both the differences between distributions from long and short utterances and the uncertainty of utterance vector representations. The results show that the proposed technique is particularly advantageous when there are a limited number of speakers in the training data.

8 CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

The goal of this thesis is to improve the accuracy of speaker verification system on utterances that are short duration, e.g., 5 seconds. Conventionally, automatic speaker verification (ASV) systems are developed and evaluated on databases that comprise of long test utterances. But in scenarios including but not limited to access control, the long duration requirement of an utterance may not be valid or applicable and verification based on short duration utterances (e.g., 5 seconds or less) will be preferred. This research then includes a comprehensive analysis of differences between long (e.g., 2.5 minutes) and short (e.g., 5 seconds) duration utterances and how they affected the state-of-the-art ASV systems. It also proposes novel utterance modelling and duration mismatch compensation algorithms to address detrimental effects caused by short duration condition. This work provides key insights to those who want to implement ASV systems on shorter duration conditions about the problems that short duration utterances will cause and possible solutions. The major contributions made by this thesis can be concluded as:

I. the proposed parallel speaker and content modelling system to have a better model for text-dependent ASV;
II. the findings that total variability model can generate inaccurate utterance representation (i-vector) that are sensitive to phonetic mismatch that arises in short utterances;
III. the proposed generalised variability model which provides better utterance representation and a complementary local acoustic variability model compared to the total variability model;

IV. the proposed speaker-phonetic vector representation of utterance to address the phonetic mismatch issue;

V. the analyses of duration mismatch in utterance vector representation space (e.g., i-vectors);

VI. the proposed duration compensation methods including twin model projection and deep neural network (DNN) based non-linear compensation methods in utterance vector representation space and a novel back-end (Twin Model GPLDA) that is tailored for the scenario with long duration enrolment and short duration test.

8.1.1 Proposal of Parallel Speaker and Content Modelling for Text-dependent Speaker Verification

In text-dependent speaker verification, there are three potential alternative hypotheses, namely, that the speaker is not the claimed speaker but the pass-phrase is right ($H_{(\bar{\chi},\not{p})}$), that the speaker is the claimed speaker but the test utterance is not the expected pass-phrase (denoted as $H_{(\chi,\bar{\not{p}})}$), and the speaker is not the claimed speaker and the test utterance is not the expected pass phrase ($H_{(\bar{\chi},\bar{\not{p}})}$). These three alternative hypotheses may be referred to as imposter-correct, target-wrong and imposter-wrong respectively. How to model those potential alternative hypotheses into one model is challenging.

As part of this thesis, a parallel speaker and content modelling system, which is introduced in Section 3.1, is proposed to address this challenge. This system uses two systems to handle the verification of the speaker's identity and the lexical content of the passphrase in parallel, with one sub-system modelling speaker identity based on the assumption that lexical content is known and the other sub-system modelling lexical content in a speaker dependent manner. The text dependent speaker verification sub-system is based on hidden Markov models and the lexical content verification system is based on models of speech segment that use a distinct Gaussian mixture model for each segment. Unlike other systems in text-dependent system,

which often only the speaker's identity is modelled, three alternative hypotheses are modelled efficiently in this model. This gives advantages when content of passphrase is used in text-dependent ASV.

Furthermore, since each pass-phrase is extremely short and phonetic coverage is limited in short duration Text-Dependent automatic speaker verification (TD ASV), the number of adapted mixtures in each model is quite limited which makes the models quite redundant. Moreover, some mixtures adapted based only on a small number of feature frames can lead to errors and removing them could help. Thus, in Section 3.2, a mixture selection method based on KL divergence is applied to refine the lexical content sub-system by making the models more discriminative. Proposed methods then are tested on part 1 of the RedDots database, which is a database targeting for TD ASV. Experimental show that the proposed combination of two sub-systems with mixture selection outperformed the baseline system by 39.8%, 51.1% and 37.3% in terms of the 'imposter_correct', 'target_wrong' and 'imposter_wrong' metrics respectively.

8.1.2 Analysis of total variability model for short duration utterances

In order to identify the effects that are caused by short duration utterance, the total variability model which is to generate utterance representation is analysed. The major advantage of using total variability model is that it generates so-called i-vector that can represent corresponding utterance with little loss of accuracy if the utterance is long (e.g., 2.5 minutes). In Section 4.1, we use the trace of covariance matrix of supervectors mapped by i-vector, which is to indicate how accurate the i-vector in terms of encoding corresponding utterance. It has been found that the accuracy drops rapidly over duration. This suggests that i-vector is not an accurate representation for short utterance. Moreover, statistics of each utterance, which are called N-vector in Section 4.1, calculated from a UBM is used as a vector representation of utterance and compared with i-vector. This experiment shows the link between the statistics and i-vector, that

statistics of short duration utterance are more disperse which caused larger uncertainty of i-vector to represent corresponding utterance.

Moreover, it has been shown in Section 5.1 that short duration utterances make corresponding i-vectors sensitive to phonetic mismatch. The analysis is taken by using phonetic i-vectors which represents the information in one utterance of corresponding phoneme. Results indicate that i-vectors from different phonemes are not uniformly distributed in the original i-vector space, and support the ideas that an i-vector is not phonetically invariant and that the output of the total variability model contains phonetic information. This would not be a problem for long duration utterances in the total variability model as the amount of information is sufficient to cover all phonetic events and the statistical patterns for each phonetic group that contain one or more phonetic events are relatively stable. Consequently, the extracted i-vector will not be biased toward a particular group and the within-class covariance is not enlarged. However, for short duration utterances, the amount of information in each group is not statistically stable. This will make the extracted i-vector biased towards some dominant groups, and therefore introduce further mismatch to scenarios where long duration utterances are served as enrolment data and short duration utterances are used as test files. This is the major reason that i-vectors are sensitive to content variation in short duration utterances.

8.1.3 Generalised variability model and the complementary local acoustic variability model

As analysed in Section 4.1, i-vectors generated by total variability model reduce accuracy for short duration utterances. In the conventional total variability model, the distribution of the latent variables is modelled as Gaussian distribution. But this may not be optimal, and it does not have any mechanism to remove phonetic variation in the i-vector space which can be severe in short utterances. Thus, in Section 4.2, a generalised variability model is developed. In the development of the algorithm, a mixture of Gaussian (MoG) prior distribution of the latent variables is used, with the assumption that each mixture is generated by a different source of

variation. Each source has its own prior distribution and each prior bears the full responsibility of modelling local cluster information (e.g., phoneme information or channel information). The latent variable distribution is then a combination of these different clusters. The posterior probability of each source can be provided by a well-trained classifier (e.g., phoneme decoder). By relaxing some of the constraints of the total variability model, the generalised variability model (GVM) can model arbitrary distribution of latent variable. Experimental results show that with a MoG prior on latent variable, better performance can be obtained and in particular greater gains, up to 15%, can be achieved when modelling short utterances.

In Section 4.3, as the uncertainty of the i-vector representation increases sharply for short duration utterances, the latent variable model in total variability model is bypassed and directly captures local acoustic variability information in the supervector space. The information in each phonetic group, referred to as local acoustic variability, is complementary to the total variability model. A Gaussian Probabilistic Linear Discriminative Analsysis (GPLDA) with block diagonal assumption on top of supervector is developed. It is showed that the model scores as mixture-wise manner, which suggests to compare information within each local acoustic group. This eventually captures local acoustic information. Following this, different weighting strategies are applied in order to take the relative reliability of local acoustic information into account. Experiments confirm that the proposed method is complementary with total variability model by fusion scores from each model.

8.1.4 Speaker-phonetic vector representation of utterance

As analysed in Section 5.1, i-vectors generated by total variability model are sensitive to phonetic mismatch which can be severe in shorter duration utterances. This sensitivity originates from the fact that i-vectors from different phonemes are not uniformly distributed and that the output of the total variability model contains phonetic information. Instead of having one single vector representation of all phonemes in one utterance, one way to remove the

phonetic variation is generating a distinct vector representation for each phonetic class in one utterance. These vectors are then called speaker-phonetic vector representation of utterance.

In the first method, this idea is realised by revising the generalised variability model. Each mixture of prior distribution (which is a MoG) modelling the prior of one phoneme is fully responsible to map phonetic information into the speaker-phonetic vector space. The second model proposed in Section 4.3 is mixtures of the total variability model (MTVM). It has a hierarchical structure in which the first layer, under each mixture component of the first layer, there is a complete total variability model with a shared latent variable. Another version of MTVM is the tied parameter version, in which the same projection matrix is shared by all phonetic supervectors. The motivation of tying parameter is that it reduces the computational load by tying the factor loadings in MTVM but it is still able to represent utterance by a number of vectors with both speaker-specific and phonetic information. These methods are tested on the NIST SRE 2010 CORE-10SEC and 8CONV-10SEC conditions. Results show that improvements can be obtained with those methods. Significant improvements are also observed when fusing with total variability model, suggesting that capturing local phonetic information is also complementary.

8.1.5 Mismatch compensation techniques for short duration

Analysis in Section 6.1 and Section 7.1 show that the vectors representing short utterances are distributed differently to those representing long utterances and this will propagate into the back-ends. In Section 6.1, by projecting utterance vector representations into three dimensions, it can be visualised that i-vectors from short utterances have larger uncertainties and are expected to have larger covariances to model distributions of i-vectors from short utterance. This means that it is may not be accurate to model those vectors from long and short utterances with identical distributions and also signifies the need to normalise the distribution mismatch caused by utterance duration. The analysis in Section 7.1 reveals that duration mismatch exists

in utterance vector representations, even though pre-processing is performed (without duration compensation), and that this needs to be considered in the design of a back-end classifier. It is shown that the histogram of i-vector magnitudes (after length normalization) estimated from long and short utterances are mismatched, signalling that utterance vector representations from short and long utterances are still sampled from different distributions. This however violates the assumption, that utterance vector representations are sampled from the same distribution, in the commonly used GPLDA back-end.

In order to address this problem, compensation techniques are proposed in Section 6.2-6.3 and Section 7.2-7.3. Specifically, a linear projection method based on the GPLDA model that model vector representations from long and short utterance separately but shared with tied latent variables is proposed in Section 6.2. This model considers both intra-speaker variability and inter-speaker variability. After the projection, values of KL divergence estimated from vector representations of long and short utterance show that duration mismatch problem is mitigated, and experimental results also confirm this. Additionally, a non-linear projection algorithm based on neural networks is also proposed to normalise duration mismatch in the utterance vector representation space by a novel duplet centre loss. In the projected space, utterance vector representations between long and short utterances are more likely to be similarly distributed. Thus, the assumption that the same distribution is shared by enrolment and test data is more likely to be satisfied in the back-end. It shows in the experiments that distances of utterance vector representations of different speakers increase while distances of utterance vector representations of the same speaker decrease. Experimental results confirm that those distances affect ability of utterance vector representations to verify speaker's identity.

In Chapter 7, Twin Model GPLDA (TM-GPLDA) is proposed. This novel back-end, which uses two different sets of parameters to model short and long utterances separately, while being connected by sharing the same speaker identity, can account for this discrepancy and generates

more reliable scores. More specifically, TM-GPLDA model uses two Gaussian distributions to model vector representations from long and short utterances, but these two distributions are not totally independent. In the model, the same latent variable is shared by both vector representations of long and short utterances, which represent the underlying linkage of two distributions. Furthermore, the TM-GPLDA can accommodate uncertainties of utterance vector representations, which considers the reliability of corresponding utterance vector representation. Experimental results in Section 7.3 show that improved performances are obtained with the new back-end.

8.2 Future perspectives

This thesis focuses on modelling and compensation techniques for short duration speaker verification under the conventional structure presented in Section 2.1. Although feature extraction is not in the scope of this research, it may also be interesting to investigate as more speaker discriminative features in the front-end are desired. As an example, deep neural networks may help to extract more speaker discriminative features. Conventionally, to extract features from speech segments, filter bank using Mel scale is designed as mentioned in Section 2.3.1. Deep neural networks with supervised learning methods can be potentially used to learn parameters to extract more speaker discriminative features as those parameters are directly minimise objective function which maximise the ability to recognise speakers.

Problems revealed in this thesis are not exhaustively solved by methods proposed in this research. For example, the first problem found in this thesis is that conventional i-vector representation of utterance is not accurate for short duration utterance. The proposed methods for this problem focus on factor analysis techniques. Other speaker embedding that is trained other than EM algorithm, such as Maximum Mutual Information (MMI) criterion may be interesting to investigate. Models proposed in this thesis are trained by point estimation and

optimised by EM algorithm. Bayesian method for parameter estimation may also be interesting in this context.

The conventional automatic speaker verification (ASV) structure mentioned in Section 2.1 is divided into several parts. Models in each part are optimised by its own criterion. But those criteria are not the same with each other. A structure that can merge those parts together and optimised by one criterion may be more reliable. An end-to-end deep neural network is one of these structures that should be interesting to investigate. It may be used to better model these systems and may be able to capture more discriminative features that would be more promising for short duration speaker verification.

A mechanism to quantify the amount of information in a given utterance would be informative for short duration speaker verification and would be useful as a metric other than overall system performance. In speaker verification, performance is the only tool to analyse how well the system works. A quantitative analysis of the information contained in an utterance reflecting speaker discriminability would be extremely useful for speaker verification. This present underlying how much information a system can use and how much has been used. This is expected to pursue in the future work.

Finally, the research in this thesis may also benefit other problem in ASV such as channel mismatch and noisy condition in ASV. Take the generalised variability model in Section 3.2 as an example. The different sources in the prior distribution of this model are not constrained with phonetic classes. Different channels or different Signal-to-Noise ratio can also serve as different sources in the prior distribution of generalised variability model. Vector representations of utterances then are expected to be less affected by channel variability or noisy levels.

APPENDIX A: EXPECTATION MAXIMIZATION ALGORITHM

The expectation maximization algorithm, or EM algorithm, is a general technique for finding maximum likelihood solutions for probabilistic models that have latent variables [144]. Latent variables are opposite to observed variables and are variables that are not directly observed but are rather inferred.

Without loss of generality, let X denote the observed data, Z the latent variables, and θ the model parameters. The likelihood to maximize is the conditional probability of X given θ given by

$$p(X|\theta) = \sum_{Z} p(X, Z|\theta) \tag{A-1}$$

where $\sum_Z p(\cdot)$ denotes the summation operator that sum over the range of Z and assuming that Z is discrete, otherwise the summation should be replaced by integral.

In many cases, directly optimizing $p(X|\theta)$ is not trivial, while optimizing $p(X, Z|\theta)$ is much easier. For this purpose, by introducing an arbitrary distribution over Z, $q(Z)$, the following decomposition holds,

$$log[p(X|\theta)] = \mathcal{L}(q, \theta) + KL\big(q(Z)||p(Z|X, \theta)\big) \tag{A-2}$$

where $KL(\cdot\,||\,\cdot)$ is the Kullback–Leibler (KL) divergence [101] between two distributions, $KL\big(q(Z)||p(Z|X, \theta)\big)$ is denoted as $KL(q||p)$ for brevity, the term $\mathcal{L}(q, \theta)$ is a functional of the distribution $q(Z)$ and a function of the parameters θ, serves as a lower bound for log-likelihood function $log[p(X|\theta)]$ and is written as

$$\mathcal{L}(q,\theta) = \sum_z q(Z) log\left[\frac{p(X,Z|\theta)}{q(Z)}\right]. \quad \text{(A-3)}$$

The KL divergence term is written as

$$KL\big(q(Z)||p(Z|X,\theta)\big) = -\sum_z q(Z) log\left[\frac{p(Z|X,\theta)}{q(Z)}\right]. \quad \text{(A-4)}$$

This must satisfy $KL(q||p) \geq 0$, with equality if and only if $q(Z) = p(Z|X,\theta)$. This decomposition is illustrated by Figure A.1.

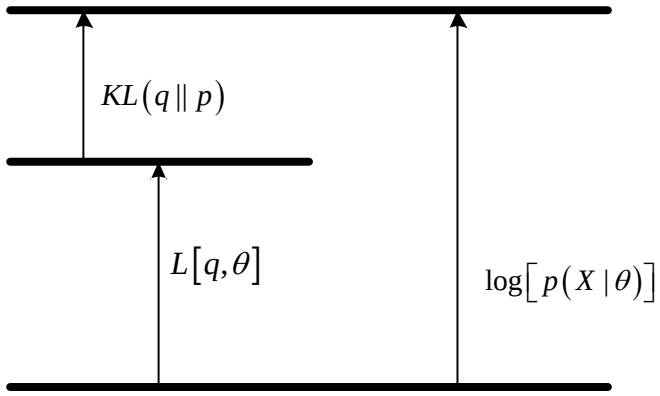


Figure A.0.1 An illustration of the likelihood decomposition equation (2-4) [50].

The EM algorithm is a two-stage iterative optimization technique for finding maximum likelihood solutions [50]. In the E-step, the parameters θ are fixed to θ^{old}. The lower bound $\mathcal{L}\big(q,\theta^{old}\big)$ can be maximized by finding the posterior probability $p(Z|X,\theta)$, and letting $q(Z) = p\big(Z|X,\theta^{old}\big)$ as $KL(q||p)$ has vanished. In the M-step, the distribution $q(Z)$ is held the same as in the previous iteration, while the lower bound can be increased by optimising it against the parameters θ. This subsequently increases the log-likelihood function as well if it is not a maximum, as $KL(q||p)$ is no longer zero. In this M-step, the lower bound $\mathcal{L}(q,\theta)$ is

$$\mathcal{L}(q,\theta) = \sum_{z} p\left(Z|X,\theta^{old}\right) log[p(X,Z|\theta)] - \sum_{z} p\left(Z|X,\theta^{old}\right) log\left[p\left(Z|X,\theta^{old}\right)\right]$$
$$= E\left[log[p(X,Z|\theta)]\right]_{\left(Z|X,\theta^{old}\right)} - const \tag{A-5}$$

where $E(\cdot)$ is the expectation operator.

APPENDIX B: EM ALGORITHM FOR GMM TRAINING

Suppose the observed data as $x = \{x_1, x_2, \dots, x_N\}$. The Gaussian mixture distribution of this data with K mixtures is represented as

$$p(x_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \tag{B-6}$$

where π_k is k^{th} mixture's mixing coefficient, μ_k and Σ_k are the k^{th} mixture's mean vector and covariance matrix, respectively. $\mathcal{N}(x_n|\mu_k, \Sigma_k)$ is a Gaussian distribution, which can be written as

$$\mathcal{N}(x_n|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} exp\left\{-\frac{1}{2}(x_n - \mu_k)(\Sigma_k)^{-1}(x_n - \mu_k)^T\right\} \tag{B-6}$$

where D is the dimensionality of the observed data. First, we introduce a K-dimensional binary random variable z which has similar meaning of Z in Section 2.4.1. There is only one element of z that is non-zero, i.e. $z_k \in \{0,1\}$ and $\sum_k z_k = 1$. A particular Z may be like $[1, 0, , \dots, 0]$, where $z_1 = 1$ and other elements are 0. The variable z is a latent variable used to indicate which mixture an observed data x_n belongs to. So z should have a lower script to indicate the observed data. The marginal distribution of z is specified by mixing coefficient, which is

$$p(z_{nk} = 1) = \pi_k \tag{B-7}$$

Because z uses a 1-of-K representation, its distribution can be written as the product

$$p(z_{nk}) = \prod_{k=1}^{K} {\pi_k}^{z_{nk}} \tag{B-8}$$

thus, the distribution of x given z can further be written as

$$p(x|z) = \prod_{k=1}^{K} \mathcal{N}(x_n|\mu_k, \Sigma_k)^{z_{nk}} \tag{B-9}$$

A graphical representation of this model is shown in Figure B.1, where z is the latent variable and x is the observed data, π, μ and Σ are parameters, denoted as $\theta = \{\pi, \mu, \Sigma\}$.

In utilising the EM algorithm as described in Section 2.4.1.1, first the expectation step should be calculated. Using Bayes's theorem, the following equation can be obtained:

$$p(z|x, \theta) \propto p(x|z, \theta)p(z|\theta) \tag{B-10}$$

Assuming that the observed data is independent and identically distributed (i.i.d), equation (2-10) can be written as

$$p(z|x, \theta) = \prod_{n=1}^{N} \frac{\prod_{k=1}^{K}[\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)]^{z_{nk}}}{\sum_{k=1}^{K}[\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)]^{z_{nk}}} \tag{B-11}$$

According to the discussion in the previous section, lower bound $\mathcal{L}(q, \theta)$ can be used to optimise parameters. As the term $\sum_z p(z|x, \theta^{old}) log[p(z|x, \theta^{old})]$ is not relevant to parameters, so it can be ignored when optimising parameters. Thus instead of using $\mathcal{L}(q, \theta)$, an auxiliary function $\mathcal{Q}(.)$ can be obtain by removing $\sum_z p(z|x, \theta^{old}) log[p(z|x, \theta^{old})]$ in $\mathcal{L}(q, \theta)$ and it is then defined by the expectation of the joint distribution over the posterior probability of the latent variable, which can be expressed as

$$\mathcal{Q}(\theta, \theta^{old}) = \sum_z p(z|x, \theta^{old}) log[p(x, z|\theta)] \tag{B-13}$$

where $p(z|x, \theta^{old})$ is given by (B-11), and

$$log[p(x, z|\theta)] = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk}\{\pi_k log[\mathcal{N}(x_n|\mu_k, \Sigma_k)]\} \tag{B-14}$$

Expanding the Gaussian distribution inside the logarithm, the updated mean and covariance parameters can be formulated as

$$\mu_k = \frac{1}{N_k}\sum_{n=1}^{N} r(z_{nk})x_n \tag{B-15}$$

and

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N} r(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T \tag{B-16}$$

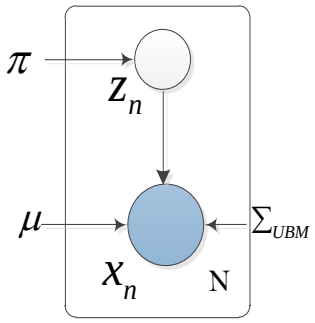


Figure B.0.1 Graphical representation of a Gaussian mixture model [50], where x_n is the observed data, z_n is the latent variables, π is the mixing coefficient, μ is the mean, Σ_{UBM} is the covariance, and N is the number of data points.

The parameters of GMM are then iteratively trained by using the formulas provided in this section.

9 REFERENCES

[1] A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: Personal identification in networked society* vol. 479: Springer Science & Business Media, 2006.

[2] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics*: Springer Science & Business Media, 2007.

[3] P. Belin, R. J. Zatorre, P. Lafaille, P. Ahad, and B. Pike, "Voice-selective areas in human auditory cortex," *Nature,* vol. 403, p. 309, 2000.

[4] H. Blank, N. Wieland, and K. von Kriegstein, "Person recognition and the brain: merging evidence from patients and healthy individuals," *Neuroscience & Biobehavioral Reviews,* vol. 47, pp. 717-734, 2014.

[5] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Processing Magazine,* vol. 32, pp. 74-99, 2015.

[6] C. R. Pernet, P. McAleer, M. Latinus, K. J. Gorgolewski, I. Charest, P. E. Bestelmeyer, *et al.*, "The human voice areas: Spatial organization and inter-individual variability in temporal and extra-temporal cortices," *Neuroimage,* vol. 119, pp. 164-174, 2015.

[7] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *2002 IEEE international conference on Acoustics, Speech, and Signal Processing (ICASSP),* , 2002, pp. IV-4072-IV-4075.

[8] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech communication,* vol. 52, pp. 12-40, 2010.

[9] T. R. Gruber, "Siri, a Virtual Personal Assistant—Bringing Intelligence to the Interface," ed: Jun, 2009.

[10] A. Nijholt, "Google home: Experience, support and re-experience of social home activities," *Information Sciences,* vol. 178, pp. 612-630, 2008.

[11] Amazon and Amazon Echo, "https://developer.amazon.com/echo."

[12] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication,* vol. 60, pp. 56-77, 2014.

[13] A. F. Martin and C. S. Greenberg, "The NIST 2010 speaker recognition evaluation," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[14] D. Garcia-Romero, "Robust speaker recognition based on latent variable models," 2012.

[15] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The 2016 Speakers in the Wild Speaker Recognition Evaluation," in *INTERSPEECH*, 2016, pp. 823-827.

[16] C. S. Greenberg, V. M. Stanford, A. F. Martin, M. Yadagiri, G. R. Doddington, J. J. Godfrey, *et al.*, "The 2012 NIST speaker recognition evaluation," in *INTERSPEECH*, 2013, pp. 1971-1975.

[17] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Rsr2015: Database for text-dependent speaker verification using multiple pass-phrases," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[18] A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, and M. W. Mason, "I-vector based speaker recognition on short utterances," in *Proceedings of the 12th Annual Conference of the International Speech Communication Association*, 2011, pp. 2341-2344.

[19] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 788-798, 2011.

[20] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in *Odyssey*, 2010, p. 14.

[21] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, D. Reynolds, E. Singer, *et al.*, "The 2016 NIST speaker recognition evaluation," presented at the INTERSPEECH, 2016.

[22] F. McGehee, "The reliability of the identification of the human voice," *The Journal of General Psychology*, vol. 17, pp. 249-271, 1937.

[23] I. Pollack, J. M. Pickett, and W. H. Sumby, "On the identification of speakers by voice," *the Journal of the Acoustical Society of America*, vol. 26, pp. 403-406, 1954.

[24] B. S. Atal, "Automatic recognition of speakers from their voices," *Proceedings of the IEEE*, vol. 64, pp. 460-475, 1976.

[25] L. G. Kersta, "Voiceprint identification," *The Journal of the Acoustical Society of America*, vol. 34, pp. 725-725, 1962.

[26] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, pp. 1437-1462, 1997.

[27] P. Belin, "Similarities in face and voice cerebral processing," *Visual Cognition*, vol. 25, pp. 658-665, 2017.

[28] P. Rose, *Forensic Speaker Identification*: CRC Press, 2003.

[29] J. J. Wolf, "Efficient acoustic parameters for speaker recognition," *The Journal of the Acoustical Society of America*, vol. 51, pp. 2044-2056, 1972.

[30] G. Doddington, "Speaker recognition based on idiolectal differences between speakers," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[31] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "Phonetic speaker recognition with support vector machines," in *Advances in Neural Information Processing Systems*, 2004, pp. 1377-1384.

[32] L. Ferrer, E. Shriberg, S. Kajarekar, and K. Sonmez, "Parameterization of prosodic feature distributions for SVM modeling in speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007.* pp. IV-233-IV-236.

[33] B. Ma, D. Zhu, R. Tong, and H. Li, "Speaker cluster based GMM tokenization for speaker recognition," in *Ninth International Conference on Spoken Language Processing*, 2006.

[34] S. Yaman, J. Pelecanos, and R. Sarikaya, "Bottleneck features for speaker recognition," in *Odyssey 2012-The Speaker and Language Recognition Workshop*, 2012.

[35] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters,* vol. 22, pp. 1671-1675, 2015.

[36] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication,* vol. 73, pp. 1-13, 2015.

[37] S. S. Stevens and J. Volkmann, "The relation of pitch to frequency: A revised scale," *The American Journal of Psychology,* vol. 53, pp. 329-353, 1940.

[38] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in *Readings in speech recognition*, ed: Elsevier, 1990, pp. 65-74.

[39] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE transactions on Computers,* vol. 100, pp. 90-93, 1974.

[40] E. Zwicker, "Subdivision of the audible frequency range into critical bands (Frequenzgruppen)," *The Journal of the Acoustical Society of America,* vol. 33, pp. 248-248, 1961.

[41] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *the Journal of the Acoustical Society of America,* vol. 87, pp. 1738-1752, 1990.

[42] R. J. Mammone, X. Zhang, and R. P. Ramachandran, "Robust speaker recognition: A feature-based approach," *IEEE Signal Processing Magazine,* vol. 13, p. 58, 1996.

[43] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 29, pp. 254-272, 1981.

[44] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE transactions on Speech and Audio Processing,* vol. 2, pp. 578-589, 1994.

[45] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. IEEE Odyssey: Speaker Lang. Recognition Workshop, Crete, Greece*, 2001.

[46] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings. (ICASSP'03).* 2003, pp. II-53.

[47] P. Mishra, "A vector quantization approach to speaker recognition," in *Proceedings of the International Conference on Innovation & Research in Technology for sustainable development (ICIRT 2012)*, 2012, p. 152.

[48] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, pp. 100-108, 1979.

[49] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 72-83, 1995.

[50] C. M. Bishop, "Pattern recognition and machine learning (information science and statistics) springer-verlag new york," *Inc. Secaucus, NJ, USA*, 2006.

[51] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, pp. 19-41, 2000.

[52] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006*, pp. I-I.

[53] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, pp. 308-311, 2006.

[54] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech & Language*, vol. 20, pp. 210-229, 2006.

[55] A. O. Hatch and A. Stolcke, "Generalized linear kernels for one-versus-all classification: application to speaker recognition," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006*, pp. V-V.

[56] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, pp. 121-167, 1998.

[57] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, vol. 14, pp. 28-29, 2005.

[58] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, pp. 308-311, 2006.

[59] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, pp. 611-622, 1999.

[60] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Ninth international conference on spoken language processing*, 2006.

[61] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of human genetics*, vol. 7, pp. 179-188, 1936.

[62] A. Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, pp. I/629-I/632 Vol. 1.

[63] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE 11th International Conference on Computer Vision, 2007*, pp. 1-8.

[64] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector Length Normalization in Speaker Recognition Systems," in *Interspeech*, 2011, pp. 249-252.

[65] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, pp. 82-97, 2012.

[66] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, pp. 293-298.

[67] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1695-1699.

[68] P. A. Torres-Carrasquillo, F. Richardson, S. Nercessian, D. Sturim, W. Campbell, Y. Gwon, *et al.*, "The MIT-LL, JHU and LRDE NIST 2016 speaker recognition evaluation system," in *Proceedings of INTERSPEECH*, 2017, pp. 1333-1337.

[69] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, "Preliminary investigation of boltzmann machine classifiers for speaker recognition," in *Odyssey 2012-The Speaker and Language Recognition Workshop*, 2012.

[70] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, "PLDA using Gaussian restricted Boltzmann machines with application to speaker verification," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[71] M. McLaren, Y. Lei, N. Scheffer, and L. Ferrer, "Application of convolutional neural networks to speaker recognition in noisy conditions," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[72] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-End Text-Dependent Speaker Verification," *arXiv preprint arXiv:1509.08062*, 2015.

[73] H. Bredin, "TristouNet: Triplet Loss for Speaker Turn Embedding," *arXiv preprint arXiv:1609.04301*, 2016.

[74] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815-823.

[75] C. Zhang and K. Koishida, "End-to-End Text-Independent Speaker Verification with Triplet Loss on Short Utterances," *Proc. INTERSPEECh 2017*, pp. 1487-1491, 2017.

[76] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *ICASSP, Calgary*, 2018.

[77] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[78] M. Hébert, "Text-dependent speaker recognition," in *Springer handbook of speech processing*, ed: Springer, 2008, pp. 743-762.

[79] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Modelling the alternative hypothesis for text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 734-738.

[80] P. Kenny, T. Stafylakis, P. Ouellet, and M. J. Alam, "JFA-based front ends for speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 1705-1709.

[81] P. Kenny, T. Stafylakis, J. Alam, and M. Kockmann, "JFA modeling with left-to-right structure and a new backend for text-dependent speaker recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4689-4693.

[82] E. Variani, X. Lei, E. McDermott, I. Lopez-Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *ICASSP*, 2014, pp. 4052-4056.

[83] P. Kenny, T. Stafylakis, J. Alam, P. Ouellet, and M. Kockmann, "Joint factor analysis for text-dependent speaker verification," in *Proc. Odyssey Speaker and Language Recognition Workshop, Joensuu, Finland*, 2014.

[84] M. J. Alam, P. Kenny, and T. Stafylakis, "Combining amplitude and phase-based features for speaker verification with short duration utterances," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[85] T. Hasan, R. Saeidi, J. H. Hansen, and D. A. van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7663-7667.

[86] A. K. Sarkar, D. Matrouf, P. M. Bousquet, and J.-F. Bonastre, "Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[87] S. Cumani, O. Plchot, and P. Laface, "On the use of i–vector posterior distributions in Probabilistic Linear Discriminant Analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 846-857, 2014.

[88] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7649-7653.

[89] L. Chen, K. A. Lee, E.-S. Chng, B. Ma, H. Li, and L. R. Dai, "Content-aware local variability vector for speaker verification with short utterance," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5485-5489.

[90] S. E. Shepstone, K. A. Lee, H. Li, Z.-H. Tan, and S. H. Jensen, "Total variability modeling using source-specific priors," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, pp. 504-517, 2016.

[91] T. Stafylakis, P. Kenny, V. Gupta, J. Alam, and M. Kockmann, "Compensation for phonetic nuisance variability in speaker recognition using DNNs." in *ISCA Odyssey*, 2016

[92] A. Larcher, P.-M. Bousquet, K. A. Lee, D. Matrouf, H. Li, and J.-F. Bonastre, "I-vectors in the context of phonetically-constrained short utterances for speaker verification," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4773-4776.

[93] N. Fatima, X. Wu, and F. T. Zheng, "Speech unit category based short utterance speaker recognition," *Computer Science and Information Systems*, vol. 9, pp. 1407-1430, 2012.

[94] N. Scheffer and Y. Lei, "Content matching for short duration speaker recognition," in *INTERSPEECH*, 2014, pp. 1317-1321.

[95] M. McLaren, Y. Lei, and L. Ferrer, "Advances in deep neural network approaches to speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015*, pp. 4814-4818.

[96] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, 1992, pp. 517-520.

[97] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, *et al.*, "Promoting robustness for speaker modeling in the community: the PRISM evaluation set," in *Proceedings of NIST 2011 workshop*, 2011.

[98] K. A. Lee, A. Larcher, G. Wang, P. Kenny, N. Brümmer, D. v. Leeuwen, *et al.*, "The RedDots data collection for speaker recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[99] T. Kinnunen and P. Rajan, "A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data," in *ICASSP*, 2013, pp. 7229-7233.

[100] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268-278, 1973.

[101] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, pp. 79-86, 1951.

[102] Y. Lei and J. H. Hansen, "Dialect classification via text-independent training and testing for arabic, spanish, and chinese," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 85-96, 2011.

[103] A. Larcher, K.-A. Lee, B. Ma, and H. Li, "RSR2015: Database for Text-Dependent Speaker Verification using Multiple Pass-Phrases," in *INTERSPEECH*, 2012, pp. 1580-1583.

[104] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, *et al.*, *The HTK book* vol. 2: Entropic Cambridge Research Laboratory Cambridge, 1997.

[105] K. A. Lee, A. Larcher, G. Wang, P. Kenny, N. Brümmer, D. van Leeuwen, *et al.*, "The RedDots Data Collection for Speaker Recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[106] T. Kinnunen, M. Sahidullah, I. Kukanov, H. Delgado, M. Todisco, A. K. Sarkar, *et al.*, "Utterance Verification for Text-Dependent Speaker Recognition: A Comparative Assessment Using the RedDots Corpus," in *Interspeech*, 2016, pp. 430-434.

[107] H. Zeinali, H. Sameti, L. Burget, J. Cernocký, N. Maghsoodi, and P. Matejka, "i-Vector/HMM Based Text-Dependent Speaker Verification System for RedDots Challenge," in *Interspeech*, 2016, pp. 440-444.

[108] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 345-354, 2005.

[109] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language recognition in ivectors space," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[110] R. Xia and Y. Liu, "Using i-vector space model for emotion recognition," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[111] M. Karafiát, L. Burget, P. Matějka, O. Glembek, and J. Černocký, "iVector-based discriminative adaptation for automatic speech recognition," in *2011 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011, pp. 152-157.

[112] M. J. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, pp. 417-428, 2000.

[113] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural computation*, vol. 11, pp. 443-482, 1999.

[114] M. Senoussaoui, P. Kenny, N. Brümmer, E. d. Villiers, and P. Dumouchel, "Mixture of PLDA models in i-vector space for gender-independent speaker recognition," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[115] J. Ma, V. Sethu, E. Ambikairajah, and K. A. Lee, "Twin Model G-PLDA for Duration Mismatch Compensation in Text-Independent Speaker Verification," *Interspeech 2016*, pp. 1853-1857, 2016.

[116] J. Ma, V. Sethu, E. Ambikairajah, and K. A. Lee, "Duration compensation of i-vectors for short duration speaker verification," *Electronics Letters*, vol. 53, pp. 405-407, 2017.

[117] M.-W. Mak, X. Pang, and J.-T. Chien, "Mixture of PLDA for Noise Robust I-Vector Speaker Verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 130-142, 2016.

[118] R. Travadi, M. V. Segbroeck, and S. S. Narayanan, "Modified-prior i-vector estimation for language identification of short duration utterances," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[119] H. Attias, "Independent factor analysis with temporally structured sources," in *Advances in neural information processing systems*, 2000, pp. 386-392.

[120] S. Cumani, O. Plchot, and P. Laface, "Probabilistic linear discriminant analysis of i-vector posterior distributions," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 7644-7648.

[121] A. K. Sarkar, D. Matrouf, P.-M. Bousquet, and J.-F. Bonastre, "Study of the Effect of I-vector Modeling on Short and Mismatch Utterance Duration for Speaker Verification," in *INTERSPEECH*, 2012, pp. 2662-2665.

[122] J. Ma, S. Irtza, K. Sriskandaraja, V. Sethu, and E. Ambikairajah, "Parallel Speaker and Content Modelling for Text-dependent Speaker Verification," in *INTERSPEECH*, 2016.

[123] A. McCree, D. Sturim, and D. Reynolds, "A new perspective on GMM subspace compensation based on PPCA and Wiener filtering," DTIC Document2011.

[124] L. Chen, K. A. Lee, B. Ma, W. Guo, H. Li, and L. R. Dai, "Local variability modeling for text-independent speaker verification," in *Proceedings of Odyssey: Speaker and Language Recognition Workshop*, 2014.

[125] L. Chen, K. A. Lee, L. R. Dai, and H. Li, "Quasi-factorial prior for i-vector extraction," *IEEE Signal Processing Letters*, vol. 22, pp. 2484-2488, 2015.

[126] C. M. Bishop, "Pattern recognition," *Machine Learning*, vol. 128, 2006.

[127] T. Hasan and J. H. Hansen, "A study on universal background model training in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 1890-1899, 2011.

[128] S. O. Sadjadi, M. Slaney, and L. Heck, "Msr identity toolbox v1. 0: A matlab toolbox for speaker-recognition research," *Speech and Language Processing Technical Committee Newsletter*, vol. 1, 2013.

[129] P. Schwarz, P. Matejka, and J. Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006, pp. I-I.

[130] N. Brümmer and E. de Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.

[131] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, pp. 993-1022, 2003.

[132] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, *et al.*, "The subspace Gaussian mixture model—A structured model for speech recognition," *Computer Speech & Language*, vol. 25, pp. 404-439, 2011.

[133] N. Brümmer and E. de Villiers, "The BOSARIS toolkit," ed, 2013.

[134] X. Feng, Y. Zhang, and J. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014 pp. 1759-1763.

[135] T. Pekhovsky, S. Novoselov, A. Sholohov, and O. Kudashev, "On autoencoders in the i-vector space for speaker recognition," in *Odyssey 2016 - the Speaker and Language Recognition Workshop*, 2016.

[136] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, *A Discriminative Feature Learning Approach for Deep Face Recognition*: Springer International Publishing, 2016.

[137] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*, 2016, pp. 499-515.

[138] J. R. Hershey and P. A. Olsen, "Approximating the Kullback-Leibler divergence between Gaussian mixture models," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, 2007, pp. IV-317-IV-320.

[139] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, *et al.*, "TensorFlow: A System for Large-Scale Machine Learning," in *OSDI*, 2016, pp. 265-283.

[140] S. Cumani, O. Plchot, and P. Laface, "On the use of i-vector posterior distributions in probabilistic linear discriminant analysis," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, pp. 846-857, 2014.

[141] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, pp. 107-145, 2001.

[142] P. Bromiley, "Products and convolutions of Gaussian probability density functions," *Tina-Vision Memo*, vol. 3, 2003.

[143] W.-w. Lin, M.-W. Mak, and J.-T. Chien, "Fast scoring for PLDA with uncertainty propagation via i-vector grouping," *Computer Speech & Language*, 2017.

[144] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (methodological)*, pp. 1-38, 1977.